Interstage Application Server

V7.0

Security System Guide

Trademarks

Trademarks of other companies are used in this user guide only to identify particular products or systems:

Product	Trademark/Registered Trademark
Microsoft, Visual Basic, Visual C++, Windows, Windows NT, Internet Information Server, and Internet Explorer	Registered trademarks of Microsoft Corporation in the U.S.A. and other countries
Sun, Solaris, Java, and other trademarks containing Java	Trademarks of Sun Microsystems, Inc., in the U.S.A. and other countries
Linux	Registered trademark of Linus Torvalds in the U.S.A. and other countries
Red Hat, RPM and all Red Hat-based trademarks and logos	Trademarks or registered trademarks of Red Hat, Inc. in the U.S.A and other countries
UNIX	Registered trademark of The Open Group in the United States and other countries
Netscape, Netscape FastTrack Server, Netscape Enterprise Server, and Netscape Navigator	Registered trademarks of Netscape Communications Corporation in the U.S.A. and other countries
CORBA, Object Management Group, OMG, OMG IDL, IIOP, Object Request Broker, and ORB	Trademarks or registered trademarks of Object Management Group Inc. in the U.S.A. and other countries
Interstage and ObjectDirector	Registered trademarks of Fujitsu Limited

This document contains technology relating to strategic products controlled by export control laws of the producing and/ or exporting countries. This document or a portion thereof should not be exported (or re-exported) without authorization from the appropriate government authorities in accordance with such laws.

Fujitsu Limited

First Edition (March 2005)
The contents of this manual may be revised without prior notice.
All Rights Reserved, Copyright © FUJITSU LIMITED 2005

Interstage Application Server Plus
Interstage Application Server Enterprise Edition

Preface

Purpose of this Document

This manual provides information on how to set up and operate a secure Interstage system.

Note

Throughout this manual Interstage Application Server is referred to as Interstage.

Who Should Read this Document?

This document is intended for users installing and operating Interstage Application Server.

It is assumed that readers of this manual have a basic knowledge of the following:

- Authentication and access control
- The Internet
- XML
- SOAP and Web services
- SOAP security extension: Basic knowledge of electronic signatures (XML electronic signature and XML partial cipher).
- Basic knowledge of the OS used

Organization of this Document

This document is organized as follows:

Part I Security Risks and Measures

- *Chapter 1 Security Risks*

  This chapter explains security risks.

- *Chapter 2 Security Measures*

  This chapter explains security measures.

Part II Authentication and Access Control

- *Chapter 3 Authentication and Access Control for the Interstage HTTP Server*

  This chapter explains how to use the authentication and access control for the Interstage HTTP Server.

Part III Firewall and Proxy Server

- *Chapter 4 HTTP Tunneling*

  This chapter explains how to use HTTP tunneling.

- *Chapter 5 HTTP Tunneling of J2EE*

  This chapter explains how to use HTTP tunneling with J2EE.

- *Chapter 6 Linkage of the Proxy*

  This chapter explains proxy linkage.

Part IV Authentication and Encrypted Communications through Support for SSL

- *Chapter 7 Setting and Use of the Interstage Certificate Environment*

  This chapter explains setting of the Interstage Certificate Environment ,and use of it.

- *Chapter 8 Setting and Use of the Certificate/Key Management Environment Using the SMEE Command*

  This chapter explains setting of the Certificate/Key Management Environment by the SMEE command, and the use of it.

- *Chapter 9 How to Use SSL with Interstage HTTP Server*

  This chapter explains how to use SSL with the Interstage HTTP Server

- *Chapter 10 How to Use SSL with the CORBA Service*

  This chapter explains how to use SSL with the CORBA Service.

- *Chapter 11 How to Use SSL with J2EE*

  This chapter explains how to use SSL with J2EE.

- *Chapter 12 How to Use SSL with Smart Repository*

  This chapter explains how to Use SSL with Smart Repository.

Part V Security Systems for Web Services (SOAP)

- *Chapter 13 Security Functions for Web Services (SOAP)*

  This chapter explains the security functions for SOAP messages.

- *Chapter 14 How to Prepare PKI Environment for Web Services (SOAP)*

  This chapter explains the key pair and certificate management environment required to use the security functions of the Web service.

- *Chapter 15 User Authentication, SOAP Digital Signature and XML Encryption for Web Services (SOAP)*

  This chapter explains how to use user authentication, SOAP electronic signatures, and XML ciphers in the Web service.

- *Chapter 16 How to Use Reliable Messaging Function for Web Services (SOAP)*

  This chapter explains how to use the transmission guarantee function in the Web service.

Part VI Security Systems for the ebXML Message Service

- *Chapter 17 How to use SSL with the ebXML Message Service*

  This chapter explains how to use SSL with the ebXML Message Service.

- *Chapter 18 How to use XML Digital Signature with ebXML Message Service*

  This chapter explains how to use the XML digital signature with the ebXML Message Service.

- *Appendix A Enhancing Security(Protecting Interstage Resources)*

  This appendix explains enhancing security in the internet environment.

- *Appendix B Authentication and Access Control for the Component Transaction Service*

  This appendix explains how to use the authentication and access control in the Component Transaction Service.

Table of Contents

Chapter 1 Security Risks

Interstage Management Console and Interstage Operation Tool 1-2
Resources to be Protected 1-2
Functions to be Protected 1-2
Resources to be Protected 1-2
Possible Security Risks to Resources 1-3
Countermeasures Against Threats 1-3
Countermeasures Against Decryption of User IDs and Passwords 1-3
Countermeasures Against Exploitation of User IDs and Passwords 1-4
Countermeasures Against Tampering of Data Recorded In Files 1-4
Countermeasures Against Exploitation of Information Recorded in Files 1-4
Countermeasures Against Damage to Files 1-4
J2EE Application 1-5
Resources to be Protected 1-5
Functions Used for Operation of J2EE Applications 1-5
Resources to be Protected 1-6
Possible Security Risks 1-7
Possible Countermeasures 1-9
Countermeasures Against Decryption of Passwords 1-9
Countermeasures Against Exploitation of Passwords 1-9
Countermeasures Against Tampering of Data Recorded in Files 1-9
Countermeasures Against Exploitation of Information Recorded in Files 1-9
Countermeasures Against Damage to Data 1-10
Countermeasures Against Damage to Files 1-10
Web Services 1-11
Database Linkage Service 1-12
Resources to be Protected 1-12
Functions to be Protected 1-12
Resources to be Protected 1-13
Possible Threats to Resources 1-15
Countermeasures Against Threats 1-16

Operations Confined to Specific Users....1-16
Periodic Backup....1-18
Use of the Security Function Provided by the Resource....1-18
OLTP Function....1-19
Resources to be Protected....1-19
Functions to be Protected....1-19
Resources to be Protected....1-20
Possible Threats to Resources....1-21
Countermeasures Against Security Risks....1-22
Countermeasures Against Decryption of Passwords....1-22
Countermeasures Against Exploitation of Passwords....1-22
Countermeasures Against Tampering of Data Recorded in the File....1-23
Countermeasures Against Exploitation of Information Recorded in Files....1-23
Countermeasures Against Damage to Data....1-23
Countermeasures Against Damage to Files....1-23
Smart Repository....1-24
Resources Requiring Security Protection....1-24
Smart Repository Functions and Resources Requiring Protection....1-24
Potential Security Threats....1-25
Threats and Security Measures....1-25
Password Encryption....1-26
Communication Data Encryption....1-26
Periodic Change of Passwords....1-26
Operation by Limited Users....1-26
Periodic Data Backup....1-27
Setting Access Rights for Files....1-27
Interstage Single Sign-on....1-28
Configuration Model....1-28
Possible Threats....1-29
Deleting, Rewriting, and Exposing Server Resources....1-29
Rewriting and Exposure of Communication Contents....1-29
User Spoofing....1-29
Authentication Server Spoofing....1-29
DoS Attack....1-29
Application Risk....1-30
Client Risk....1-30
Information Leakage Threat....1-30
Security Measures....1-30
Protecting the Authentication Infrastructure Setup File and Business System Setup File....1-30

Setting Access Permission for Operating Resources....1-30
Protecting Communication Contents....1-31
Confirming the Authentication Server....1-31
Countermeasures Against Password Attacks....1-31
Operating and Managing a Business Server....1-33
Application Programming....1-33
Applying Patches....1-34
Messages Displayed on the Web Browser....1-34
Multi Server Management....1-35
Configuration Model....1-36
Resources to be Protected....1-38
Functions to be Protected....1-38
Resources to be Protected....1-38
Possible Security Risks to Resources....1-38
Threat Prevention....1-39
Countermeasures Against Decryption of User IDs and Passwords....1-39
Countermeasures Against Exploitation of User IDs and Passwords....1-39
Countermeasures Against Tampering of Data Recorded In Files....1-39
Countermeasures Against Exploitation of Information Recorded in Files....1-40
Countermeasures Against Damage to Files....1-40
Configuration Management Function....1-41
Configuration Management Function Usage Model....1-41
Resources to be Protected....1-41
Functions to be Protected....1-41
Resources to be Protected....1-42
Possible Security Risks to Resources....1-42
Threat Prevention....1-42
Countermeasures Against Overwriting Information Recorded in Files....1-43
Countermeasures Against Exploiting Information Recorded in Files....1-43
Countermeasures Against File Corruption....1-43

Chapter 2 Security Measures

Common Security Measures....2-2
Notes on User Accounts....2-2
Backup....2-2
Notes on Interstage Installation Resources....2-2
Security Measures for Interstage Operation Tool....2-3
Notes on User Accounts....2-3
Notes on the Permissions of the Environment Definition File....2-3
Notes on Communication Data....2-3

Security Measures for Operation of the Web Server (Interstage HTTP Server) 2-4
Notes When Making Access 2-4
Notes on Communication Data 2-4
Threats of Denial of Service Attacks (DoS) 2-4
Leakage of Password Information 2-5
Unauthorized Access to Resource Files 2-5
Risk of Exploiting the HTTP TRACE Method 2-6
Threat that the UNIX account name will be discovered 2-7
Security Measures for Operation of the Web Server (InfoProvider Pro) 2-9
Notes on Permissions of Contents 2-9
Notes on the Permissions of the Environment Definition File 2-9
Notes on User Authentication 2-9
Security Measures for the Servlet Service 2-10
Notes on the Use of Sessions 2-10
Notes on Web Application Development 2-10
Notes on Deployment of Web Applications 2-10
Notes on the Root Directory of the Web Application 2-10
Notes on Communication Data 2-11
Security Measures for the EJB Service 2-12
Resources to be Protected 2-12
Resources to be Protected 2-12
Possible Threats to Resources 2-13
Countermeasures Against Threats 2-13
Confining Operation to Specific Users 2-13
Periodic Backup 2-14
SSL Encryption 2-14
Security Measures for J2EE Deployment Tool 2-15
Unauthorized Access to Resource Files 2-15
Security Measures for the J2EE Resource Access Definition 2-16
Leakage of Password Information 2-16
Security Measures for Interstage JMS 2-17
Unauthorized Access to Resource Files 2-17
Security Measures for CORBA Service 2-18
Unauthorized Access to Resource Files 2-18
Notes on Communication Data 2-19
Notes on the Port Number used by CORBA Service 2-19
Notes on Creation and Operation of Java Applets 2-19
About Authorization Settings 2-19
About Errors and Exceptions 2-19

Security Measures for Portable-ORB 2-20
Unauthorized Access to Resource Files 2-20
Notes on Communication Data 2-20
Notes on Creation and Operation of Java Applet 2-21
About Authorization Settings 2-21
About Errors and Exceptions 2-21
Security Measures for Event Service 2-22
Unauthorized Access to Resource Files 2-22
Security Measures for IJServer Operation 2-23
Unauthorized Access to Resource Files 2-23
Notes on IJServer Execution 2-23
Security Measures Concerning Operation of Smart Repository 2-24
About Operation 2-24
Blocking External Access 2-24
Restriction of Services 2-24
Notes on Accessing the Smart Repository Server 2-24
Security Measures for Fujitsu Enabler 2-25
Account Used for Fujitsu Enabler 2-25
Measures for Multi server Management 2-26
Security Role Settings 2-26
Measures for Configuration Manager 2-27
Illegal Access to Resource Files 2-27

Chapter 3 Authentication and Access Control for the Interstage HTTP Server

Types of Authentication 3-2
User Authentication (Basic Authentication) 3-2
IP Access Control 3-3
Online Collation 3-4
Setting the User Authentication 3-5
Registering a User Password 3-5
Editing the Environment Definition File 3-6
Relating Directives 3-7
AuthName 3-8
AuthType 3-8
AuthUserFile 3-9
<Directory> 3-9
Require 3-10
Setting the IP Access Control 3-11

Relating Directives 3-12
Allow 3-12
Deny 3-13
<Directory> 3-14
Order 3-14
Setting the Online Collation Function 3-15
Operation without Using SSL 3-16
Configuration Procedure 1 3-16
Operation Using SSL 3-16
Configuration Procedure 2 (when Interstage HTTP Server and Smart Repository are on the same system) 3-16
Configuration Procedure 3 (when Interstage HTTP Server and Smart Repository are on different systems) 3-17
Setting the Directory Server Environment 3-18
Preparing the Directory Server 3-18
Creating Entries 3-18
Set the Interstage HTTP Server Environment Definition File 3-20
Setting 1: Operation without Using SSL 3-21
Setting 2: Operation Using the SSL (setting for using an Interstage certificate environment or for using SSL configured on Smart Repository) 3-23
Setting 3: Operation Using the SSL (setting for using a certificate/key management environment configured with the SMEE commands) 3-25
Relating Directives 3-28
AddModule 3-29
AuthLDAPbasedn 3-29
AuthLDAPBindDN 3-30
AuthLDAPBindPassword 3-31
AuthLDAPCertPath 3-31
AuthLDAPEnabled 3-32
AuthLDAPHost 3-32
AuthLDAPPort 3-33
AuthLDAPSecure 3-33
AuthLDAPSlotPath 3-34
AuthLDAPTknLbl 3-35
AuthLDAPTknPwd 3-35
AuthName 3-36
AuthType 3-36
<Directory> 3-37
Group 3-37
LoadModule 3-38
Require 3-39

ServerRoot....3-40
User....3-41

Chapter 4 HTTP Tunneling

HTTP Data Communication Using HTTP Tunneling....4-2
HTTP Tunneling Mechanism....4-2
Developing the CORBA Application....4-3
Constructing the HTTP Tunneling Environment (Constructing the HTTP Gateway Environment)....4-3
Operating HTTP Tunneling....4-3
HTTP Tunneling Setup....4-4
Overview....4-4
Setting up the Web Server Environment....4-4
Establishing HTTP-IIOP Gateway....4-4
Writing HTML....4-7
Setting up HTTP Tunneling....4-8
Application Other than the Java Applet....4-10
Java Applets....4-10
Setting to be Made When an HTTP Proxy Server is to be Used....4-13

Chapter 5 HTTP Tunneling of J2EE

Use of HTTP Tunneling in J2EE Application Client....5-2
Method for Using HTTP Tunneling with IJServer (Contains Web Applications Only)....5-5
Method for Using HTTP Tunneling with Java Applets....5-6

Chapter 6 Linkage of the Proxy

Linkage of the Proxy and SOAP Service....6-2

Chapter 7 Setting and Use of the Interstage Certificate Environment

Certificates and Private Keys....7-2
What are the Certificates and the Private Keys?....7-2
CA (Certification Authority)....7-4
Configuring Environments....7-5
Using a CSR (Certificate Signing Request)....7-5
Using PKCS#12 Data....7-6
Setting up Access Permissions in the Interstage Certificate Environment....7-6
Configuring the Interstage Certificate Environment with CSR....7-8
Configuring an Interstage Certificate Environment and Creating a Certificate Signing Request (CSR)....7-9
Requesting Certificate Issuance....7-10
Registering Certificates and CRL....7-10

Registering the CA Certificate....7-11
Registering a Site Certificate....7-11
Registering the Certificate of Another Reliable Site....7-12
Registering a CRL....7-12
Configuring the Interstage Certificate Environment with PKCS#12....7-13
Configuring the Interstage Certificate Environment....7-13
Registering PKCS#12 Data, Certificates, and CRLs....7-14
Registering the CA Certificate....7-14
Importing the PKCS#12 data....7-15
Registering the Certificate of Another Reliable Site....7-15
Registering a CRL....7-16
Configuring Certificate Settings....7-17
Defining the Use of Certificates....7-17
Setting Up Various Service Environments....7-17
Certificate Management....7-19
Updating a Certificate (Before Expiration)....7-19
If a New Certificate and CRL are Obtained....7-20
Verifying Operation using a Test Site Certificate before System Operation Begins....7-20
Deleting a Certificate....7-20
Making a PKCS#12 Data Backup and Restoring from this Backup....7-20

Chapter 8 Setting and Use of the Certificate/Key Management Environment Using the SMEE Command

SSL Libraries Used with the Certificate/Key Management Environment....8-2
Types of SMEE Libraries....8-2
Certificate/Key Management Environment....8-3
Environment Setting for Certificate/Key Management Environment....8-5
Creating a Certificate/Key Management Environment....8-7
Creating a Private Key and Acquiring a Certificate....8-9
Registering the Certificate and CRL....8-10
Operating the Client Certificate....8-11
Obtaining the Client Certificate....8-12
Registering the Client Certificate....8-12
Resource Registration....8-12
Management of a Certificate/Key Management Environment....8-14

Chapter 9 How to Use SSL with Interstage HTTP Server

Setting SSL for Interstage Certificate Environments....9-2
Setting SSL for Certificate/Key Management Environments Configured with the SMEE Commands....9-3

Registering the User PIN 9-3
Setting up the Environment Definition File 9-4
General Operation of SSL 9-4
SSL Operation Using the Virtual Host Function 9-6
Relating Directives 9-11
AddModule 9-12
Alias 9-12
CustomLog 9-13
DocumentRoot 9-14
ErrorLog 9-15
Group 9-16
Listen 9-17
LogFormat 9-17
Port 9-19
ScriptAlias 9-19
ServerAdmin 9-20
ServerName 9-20
ServerRoot 9-21
SetEnvIf 9-22
SSLCertName 9-22
SSLCICACertName 9-23
SSLCipherSuite 9-24
SSLEnvDir 9-26
SSLExec 9-26
SSLSlotDir 9-27
SSLTokenLabel 9-27
SSLUserPINFile 9-28
SSLVerifyClient 9-29
SSLVersion 9-30
User 9-31
<VirtualHost> 9-32

Chapter 10 How to Use SSL with the CORBA Service

SSL Linkage of the CORBA Service 10-3
Mechanism of the SSL Linkage Function 10-3
Developing the CORBA Application 10-3
Constructing SSL Linkage Environment 10-4
Acquiring and Registering Certificates (for both the Server and Client) 10-4
Setting and Registering the SSL Environment with the CORBA Service (for both the Server and Client) 10-4

Setting the SSL Information in the CORBA Application (Server Application Only) 10-4
Operating the SSL Linkage 10-5
SSL Linkage in the IPv6 Environment 10-5
CORBA Server Environment Setup 10-6
Specifying the Addition of SSL Information at Definition of Object Reference 10-6
SSL Environment Setup in Client 10-7
Defining a Private Key/Certificate in CORBA Service 10-7
Defining Private Key/Certificate 10-7
Editing config File 10-8
Environment Setup for Event Service 10-9
For Static Generation and Operation 10-9
For Dynamic Generation and Operation (for Environment Setting using the Interstage Integration Command) 10-9
For Dynamic Generation and Operation (for Environment Setting using the Event Service Operation Command) 10-10

Chapter 11 How to Use SSL with J2EE

Environment Setup for Servlet Service 11-2
Environment Setting for EJB Service 11-3
Environment Setting for Interstage JMS 11-4

Chapter 12 Using SSL for Smart Repository

Environment Setup for Using SSL between Smart Repository Client and Server 12-3
Environment Setup for Using SSL between Master and Slave in Smart Repository Replication Operation Mode 12-4

Chapter 13 Security Functions for Web Services (SOAP)

Digital Signature Function 13-2
Encryption Function of SOAP Messages 13-3
Reliable Messaging Function and Non-repudiation Function 13-4
Attachment Function of the User ID/Password to SOAP Messages 13-5
Communication via the Proxy 13-6

Chapter 14 How to Prepare PKI Environment for Web Services (SOAP)

Configuring a Certificate Environment on the Server System 14-2
Using an Interstage Certificate Environment 14-2
Relations between Certificate Environment and Application Operation 14-4
Configuring an Old Certificate Environment or Client Certificate Environment 14-5
Constructing a Key Pair/Certificate Management Environment 14-7

Constructing a Key Pair/Certificate Management Environment 14-7
Environment Construction when a Private-key is needed 14-8
Environment Construction when a Private-key is not Needed 14-11
Using a CORBA/SOAP Gateway 14-15
Using a CORBA/SOAP Server Gateway 14-15
Using a CORBA/SOAP Client Gateway 14-15

Chapter 15 User Authentication, SOAP Digital Signature and XML Encryption for Web Services (SOAP)

Setting User Authentication for SOAP Messages 15-2
Appending a User Name and a Password to a SOAP Message 15-2
Implementing an Application to Append a User Name and a Password 15-2
Append a User Name and a Password 15-2
User Authentication for SOAP Messages 15-4
Implementing an Application that Performs User Authentication on SOAP Messages 15-4
Setting User Information 15-4
Setting User Authentication Information 15-6
Settings for the SOAP Digital Signature 15-9
Generating a SOAP Digital Signature 15-9
Implementing an Application that Attaches a SOAP Digital Signature 15-9
Preparing a Private-key 15-10
Settings for the Generation of the SOAP Digital Signature 15-10
Verifying the SOAP Digital Signature for SOAP Messages 15-13
Implementing an Application that Verifies the SOAP Digital Signature 15-13
Preparing a Site Certificate and Certification Authority Certificate 15-13
Settings for the SOAP Digital Signature Verification 15-13
Settings for the XML Encryption 15-16
Encrypting SOAP Messages Using the XML Encryption 15-16
Implementing an Application that Performs Encryption Using the XML Encryption 15-16
Preparing a Site Certificate 15-16
Settings for Encryption Using the XML Encryption 15-16
Decrypting SOAP Messages Using the XML Encryption 15-20
Implementing an Application that Performs Decryption Using the XML Encryption 15-20
Preparing a Private Key 15-20
Settings for Decryption Using the XML Encryption 15-21
Fault Codes 15-23
Supported Algorithms 15-25

Chapter 16 How to Use Reliable Messaging Function for Web Services (SOAP)

PUSH Model (Receiving Messages by the Server System) 16-2
Preparing a Key Pair and Public Key Used by the Receiver Server 16-2
Deploying the Receiver Application 16-3
Preparing a Key Pair and Public Key Used by the Sender Client 16-6
Deploying the Sender Application 16-6
PULL Model (Receiving Messages by the Client System) 16-9
Preparing a Key Pair and Public Key Used by the Sender Server 16-9
Deploying the Sender Application 16-10
Preparing a Key Pair and Public Key Used by the Receiver Client 16-12
Deploying the Receiver Application 16-12

Chapter 17 How to use SSL with the ebXML Message Service

Chapter 18 How to use XML Digital Signature with ebXML Message Service

Appendix A Enhancing Security (Protecting Interstage Resources)

Protecting Interstage Resources A-2
Environment Setup for Interstage Resources Protection A-4
CORBA Service A-4
Component Transaction Service A-9
Generating an Extended System A-10
Generating an Interstage System Definition File A-10
Registering the Interstage System Definition File A-10
Initializing Interstage A-10
Executing the Security-Enhancing Environment Setup Command A-10
Notes A-11
Database Linkage Service A-15
EJB Service A-16
Environment Construction Just After Installation A-16
Environment Construction After Constructing an Environment for EJB Service Job Operation (Default System Only) A-17
Environment Construction After Constructing an Environment for EJB Job Operation (Multi-System) A-17
EJB Service Operation A-18
Details of the ejbchangemode Command A-19
Details of the Output Messages A-21
Interstage JMS A-25
Environment Construction A-25
Notes A-28
Interstage Operation Tool A-28

Appendix B Authentication and Access Control for the Component Transaction Service

User Authentication B-2
User Authentication with Authentication Objects B-2
User Authentication with Web Server Functions B-3
Access Control B-4
Security Design (User Authentication and Access Control) B-5
User Authentication B-5
InfoDirectory Server's Operational Design B-5
Operational Design of the Authentication Object B-5
InfoDirectory User Registration B-6
Handling Authentication Requests from Clients B-7
Access Control B-7
InfoDirectory Server Operational Design B-7
Selecting an Access Control Item B-7
Registering Access Permissions for Access Control Items in InfoDirectory B-8
Definitions for Access Controlled IDL B-11
WorkUnit Definition B-12
Linking the Web Server Online Access Management Function and Access Control Function B-12
Using Security Functions B-14
Installing InfoDirectory B-14
Registering InfoDirectory Entries B-14
Specifying Manager DN B-14
Linkage to the Web Server Online Reference Function B-15
Customizing Web Server Definitions B-15
Customizing HTML Page Editing Service Definitions B-15
Running Interstage when using InfoDirectory B-15
Synchronizing Information during Operation B-15
Tracing Authentication Object Log B-15

Index

Part I

Security Risks and Measures

If the system security is violated, unauthorized access by malicious attackers can cause interference and unauthorized use of system operation as well as information leakage.

This part explains the threat of security violation by the attackers and the measures to be against them when constructing a system in a network environment using the Interstage Application Server.

Chapter 1

Security Risks

This chapter explains the resources to be protected (protection target resources), possible threats to the protection target resources, and measures to be taken against the individual threats. The chapter uses representative operation models of the Interstage Application Server to explain these.

Interstage Management Console and Interstage Operation Tool

The Interstage Management Console and the Interstage Operation Tool can be used with the following products:

- Interstage Application Server Enterprise Edition
- Interstage Application Server Standard Edition
- Interstage Application Server Web-J Edition (Interstage Management Console only)
- Interstage Application Server Plus

This section gives an overview of possible security risks in the general operating environment of the Interstage Management Console and the Interstage Operation Tool.

Resources to be Protected

This section describes the resources to be protected when the Interstage Management Console and the Interstage Operation Tool are used.

Functions to be Protected

The following functions and procedures should be protected:

- User authentication
- Connection to Web server
- Interstage Management Console
- Interstage Operation Tool environment setup

Resources to be Protected

The following resources should be secured in order to protect their corresponding function:

Table 1-1 Resources to be Protected

Function	Resource
User authentication	User ID and password used for authentication
Connection to Web server (When Interstage HTTP Server is used)	Definition file for Interstage HTTP Server
Connection to Web server (When InfoProvider Pro is used)	Definition file for the InfoProvider Pro
Interstage Operation Tool environment setup	Definition file for the Interstage Operation Tool

Possible Security Risks to Resources

The following describes possible security threats during operation of the Interstage Management Console and the Interstage Operation Tool.

Table 1-2 Possible Security Risks to Resources

Resource	Possible threat
User authentication	- Exploitation of user IDs and passwords - Decryption of user IDs and passwords
Definition file for Interstage HTTP Server	- Tampering with data recorded in the file - Exploitation of information recorded in files - Damage to files
Definition file for the Interstage Operation Tool	- Tampering with data recorded in the file - Exploitation of information recorded in files - Damage to files

Countermeasures Against Threats

The following table lists possible countermeasures against security risks.

Table 1-3 Countermeasures

Possible threat	Countermeasures
Decryption of User IDs and Passwords	- Constraint on the user ID and password
Exploitation of user IDs and passwords	- Setting the expiration date of the user ID and password
Tampering of data recorded in the file	- Setting access permissions on the file storing the information - Periodic data backup
Exploitation of information recorded in files	- Setting access permissions on the file storing the information

Countermeasures Against Decryption of User IDs and Passwords

In an environment open to the public like the Internet, user IDs or passwords may be decrypted on their transmission route. The Interstage Management Console and the Interstage Operation Tool implements encryption of user IDs and passwords, but it is still possible for them to be decrypted. To minimize this risk, set expiration dates on user IDs and passwords and change them periodically.

Countermeasures Against Exploitation of User IDs and Passwords

In an environment open to limited users like an intranet, it is not likely that user IDs and passwords will be decrypted. Such an environment is often the management base of user ID and password information, and the information of user IDs and passwords is often saved in a file. If this file is accessible by unauthorized users, there is a high risk of exploitation of the user ID and password information. An effective countermeasure against this threat is to set appropriate access permissions to files storing user ID and password information.

Countermeasures Against Tampering of Data Recorded In Files

To use the Interstage Management Console and the Interstage Operation Tool, the Interstage HTTP Server environment definition file is required. If the information in this file is illicitly tampered with, it may disable the Interstage Management Console and the Interstage Operation Tool and cause various problems. An effective countermeasure against this threat is to set appropriate access permissions on this file. For Solaris OE system or Linux system, refer to Enhancing Security (Protecting Interstage Resources) in Appendix A.

Periodic backups are also effective. For information about backups, refer to Maintenance (Resource Backup) in the Interstage Operator's Guide.

Countermeasures Against Exploitation of Information Recorded in Files

There are files storing information necessary for operation of the Interstage Management Console and the Interstage Operation Tool. The contents of these files are also a part of resources, and it is important to prevent exploitation of them. To cope with the threat of exploitation of information, it is effective to set appropriate access permissions on these files. For Solaris OE system or Linux system, refer to Enhancing Security (Protecting Interstage Resources) in Appendix A.

Countermeasures Against Damage to Files

In the environment of the Interstage Management Console and the Interstage Operation Tool, there are important files like the environment definition file. If information in these files is illicitly tampered with, it may disable the Interstage Operation Tool and cause various problems. An effective countermeasure against this threat is to set appropriate access permissions on these files. For Solaris OE system or Linux system, refer to Enhancing Security (Protecting Interstage Resources) in Appendix A.

Periodic backups are also effective. For information about backups, refer to Maintenance (Resource Backup) in the Operator's Guide.

J2EE Application

This section gives an overview of security risks in J2EE applications.

Generally, a J2EE application performs operations with client programs using various components. The client program of a J2EE application is sometimes executed as an independent Java program and sometimes via a Web browser. When it is executed via a Web browser, a Web server mediates the operation. The Web server is generally located in a Demilitarized Zone (DMZ) so that accesses to the Web browser and intranet area go through a firewall.

Resources to be Protected

This section describes the resources to be protected when a general J2EE application is used.

Functions Used for Operation of J2EE Applications

The following functions require security for operation of a general J2EE application:

- User authentication
- Connection to Web server
- Invocation of Servlet (when an V5.0.1 or earlier version of Servlet service is used)
- Invocation of EJB (during operation in old version compatible environments)
- Invocation of Servlet and EJB
- Reading data from a database
- Writing data to a database
- Operating environment setup for Web Server
- Execution environment setup for Servlet (when an V5.0.1 or earlier version of Servlet service is used)
- Execution environment setup for EJB (during operation in old version compatible environments)
- Execution environment setup for Servlet and EJB
- Deployment of a J2EE application

Resources to be Protected

The following table lists the resources that are used when the corresponding function available for a J2EE application is used. If high security is required, it is best to protect these resources.

Table 1-4 Resources to be Protected

Function	Resource to be protected
User authentication	Password used for authentication
Connection to Web server(When InfoProvider Pro is used)	Log files for InfoProvider Pro Access log Error log Access log of the extended CGI System log
Connection to Web server(When Interstage HTTP Server is used)	Log files for Interstage HTTP Server Access log Error log
Invocation of Servlet and EJB	IJServer log file Container log Container information log
Invocation of Servlet (when an V5.0.1 or earlier version of Servlet service is used)	Log file for JServlet environment Log file for Servlet service Servlet gateway log file Activation log file of the Servlet container Log file for standard output/error output of the Servlet container Log file of Servlet Log file of the Servlet container
Invocation of EJB (during operation in old version compatible environments)	Log file for EJB environment Log file for standard output output of EJB application Log file for standard error output of EJB application
Reading data from a database	Log in the database Data in the database
Writing data to a database	Log in the database Data in the database
Operating environment setup for Web Server(When Interstage HTTP Server is used)	Definition file for the Interstage HTTP Server
Operating environment setup for Web Server(When InfoProvider Pro is used)	Definition file for the InfoProvider Pro

Function	Resource to be protected
Execution environment setup for Servlet and EJB	IJServer environment definition file
Execution environment setup for Servlet (when an V5.0.1 or earlier version of Servlet service is used)	Definition file for Servlet execution environment JServlet environment definition file Servlet Gateway environment definition file Servlet Container environment definition file Web application environment definition file(deployment descriptor)
Execution environment setup for EJB (during operation in old version compatible environments)	Definition file for EJB execution environment EJB operation using WorkUnit: WorkUnit definition file EJB operation without WorkUnit: EB definition file Deployment descriptor file CMP definition file DB access environment definition file Definition file for the run-as function
Deployment of a J2EE application	ear, war, jar, and rar deployment files

Possible Security Risks

The following describes the possible security threats posed to resources to be protected during the operation of a J2EE application.

Table 1-5 Possible Security Risks

Resource to be protected	Possible threat
Password used for authentication	Decryption of passwords Exploitation of passwords
Log files for Interstage HTTP Server	Tampering of data recorded in the file Exploitation of information recorded in files Damage to files
Log files for InfoProvider Pro	Tampering of data recorded in the file Exploitation of information recorded in files Damage to files

Resource to be protected	Possible threat
IJServer log file	Tampering of data recorded in the file Exploitation of information recorded in files Damage to files
Log file for JServlet environment	Tampering of data recorded in the file Exploitation of information recorded in files Damage to files
Log file for EJB environment	Tampering of data recorded in the file Exploitation of information recorded in files Damage to files
Log in the database	Tampering of data recorded Exploitation of information recorded Damage to data
Data in the database	Tampering of data recorded Exploitation of information recorded Damage to data
Definition file for the Interstage HTTP Server	Tampering of data recorded in the file Exploitation of information recorded in files Damage to files
Definition file for the InfoProvider Pro	Tampering of data recorded in the file Exploitation of information recorded in files Damage to files
IJServer environment definition file	Tampering of data recorded in the file Exploitation of information recorded in files Damage to files
Definition file for Servlet execution environment	Tampering of data recorded in the file Exploitation of information recorded in files Damage to files
Definition file for EJB execution environment	Tampering of data recorded in the file Exploitation of information recorded in files Damage to files
ear, war, jar, and rar deployment files	Damage to files

Possible Countermeasures

The following outlines possible countermeasures against security risks. For further details, refer to the descriptions for each component.

Table 1-6 Countermeasures

Possible threat	Countermeasures
Decryption of passwords	Encryption of passwords
Exploitation of passwords	Setting access permissions of the file storing the password information
Tampering of data recorded in the file	Setting access permissions on the file storing the information Periodic data backup
Exploitation of information recorded in files	Setting access permissions on the file storing the information
Damage to data	Periodic data backup
Damage to files	Setting access permissions on the file

Countermeasures Against Decryption of Passwords

In an environment open to the public like the Internet, passwords may be decrypted on their transmission route. You can minimize this risk by encrypting passwords. Using the https protocol via a Web browser is an example of this measure.

Countermeasures Against Exploitation of Passwords

In an environment open to limited users like an intranet, it is not likely that passwords will be decrypted. Such an environment may be the management base of the passwords, and password information is often saved in a file. If this file is accessible by unauthorized users, there is a high risk of exploitation of the information in the file. An effective countermeasure against this threat is to set appropriate access permissions on this type of file.

Countermeasures Against Tampering of Data Recorded in Files

There are environment definition files and other such files in the operating environment of a J2EE application. If the information in these files is illicitly tampered with, it may disable a J2EE application and cause various problems. An effective countermeasure against this threat is to set appropriate access permissions on these files. Periodic backups in preparation for tampering is also an effective measure.

Countermeasures Against Exploitation of Information Recorded in Files

There are files storing information necessary for operation of a J2EE application. The contents of these files are also a part of resources, it is important to prevent their exploitation by setting appropriate access permissions.

Countermeasures Against Damage to Data

There are some J2EE applications that use databases. For this type of application, the data stored in those databases should also be protected. In addition to the security function that the database itself has, periodic data backup is an effective countermeasure against damage to data.

Countermeasures Against Damage to Files

There are required files in the operating environment of a J2EE application. If these files are damaged for some reason, the J2EE application cannot operate. An effective countermeasure against this threat is to set appropriate access permissions on these files.

Web Services

Web services can be used with the following products:

- Interstage Application Server Enterprise Edition
- Interstage Application Server Standard Edition
- Interstage Application Server Plus

For information on security risks to web services, refer to Security Systems for Web Services (SOAP) in Part IV.

Database Linkage Service

The Database Linkage Service can be used with the following products:

- Interstage Application Server Enterprise Edition
- Interstage Application Server Plus

This section gives an outline of security risks in an operating mode where the database linkage service is used.

The database linkage function is required for the use of the distributed transaction function of a J2EE application or the global transaction function (global transaction linkage) of the OLTP function.

Resources to be Protected

This section describes the security risks when the database linkage service is used.

Functions to be Protected

The following functions and procedures are to be protected:

- OTS system environment setup
- Registration and deletion of a resource definition file
- Creation and deletion of a program for XA linkage
- Creation and deletion of a resource control program for OTS
- OTS system operation
- Operation of the resource control program for OTS
- Operation of the resource control program for JTS
- Application operation
- Manipulation of transaction
- Resource access

Resources to be Protected

The following table lists the resources used when the database linkage service is used. If high security is required, it is desirable to protect these resources.

Table 1-7 Resources to be Protected

Function	Resource to be protected
OTS system environment setup	Folder storing the OTS system information Transaction log file Folder storing the trace log
Registration and deletion of a resource definition file	Repository storing the resource definitions
Creation and deletion of a program for XA linkage	Program for XA linkage
Creation and deletion of a resource control program for OTS	Resource control program
OTS system operation	Folder storing the OTS system information Transaction log file Folder storing the trace log
Operation of the resource control program for OTS	Repository storing the resource definitions Resource access information
Operation of the resource control program for JTS	Repository storing the resource definitions RMP property file Resource access information Folder storing the trace log
Application operation	Repository storing the resource definitions Resource access information Folder storing the trace log
Manipulation of transaction	Transaction log file Folder storing the trace log
Resource access	Resource access information Folder storing the trace log

The following describes the locations of the resources to be protected:

Windows

- Folder storing the OTS system information
  Folder where the database linkage service is installed: \etc folder
- Transaction log file
  Transaction log file that was specified when the OTS system was created
- Folder storing the trace log
  Folder where the database linkage service is installed: \var folder
- Repository storing the resource definitions
  Folder where the database linkage service is installed: \etc\repository folder
- RMP property file
  Folder where the database linkage service is installed: \etc\RMP.properties file
- Resource access information
  Folder where the database linkage service is installed: \etc\repository folder

Solaris OE Linux

- Folder storing the OTS system information
  Folder where the database linkage service is installed: /etc folder
- Transaction log file
  Transaction log file that was specified when the OTS system was created
- Folder storing the trace log
  Folder where the database linkage service is installed: /var folder
- Repository storing the resource definitions
  Folder where the database linkage service is installed: /etc/repository folder
- RMP property file
  Folder where the database linkage service is installed: /etc/RMP.properties file
- Resource access information
  Folder where the database linkage service is installed: /etc/repository folder

Possible Threats to Resources

The following describes the possible security risks to the database linkage service:

Table 1-8 Possible Security Risks

Resource to be protected	Possible threat
Folder storing the OTS system information	Tampering of information Exploitation of information Damage to data Damage to file
Transaction log file	Tampering of information Exploitation of information Damage to data Damage to file
Folder storing the trace log	Tampering of information Exploitation of information Damage to data Damage to file
Repository storing the resource definitions	Tampering of information Exploitation of information Damage to data Damage to file
RMP property file	Tampering of information Exploitation of information Damage to data Damage to file
Resource access information	Decryption of passwords Exploitation of passwords

Countermeasures Against Threats

For the database linkage service, the following are effective measures against security invasion.

- Operation confined to specified users
- Periodic backup
- Use of the security function provided by the resource

Operations Confined to Specific Users

Operations confined to specific users can be effectively secured against the following threats:

- Tampering of information
- Exploitation of information
- Damage to data
- Damage to file
- Exploitation of passwords

Operations confined to specific users can be effectively secured by the following three procedures:

- Restraint on Services
- Construction of Environment by Specific Users
- Changing Access Permissions of Protected Resources

Restraint on Services

By restricting remotely accessible services (such as telnet and ftp) on nodes where Interstage is operating, you can prevent unauthorized accesses. This measure is effective against unauthorized accesses made through networks.

For details about how to restrict remotely accessible services, refer to the manual for each platform.

Construction of Environment by Specific Users

By restricting the operation of the entire system to specific users, you can prevent tampering of information. For the database linkage service, specific users shall be selected as described below for each platform.

Administrator (Administrator user)

root (Superuser)

Using only the authorization of the selected users, start construction of the environment and operation of the database linkage service. If the environment is already established, do the following according to the functions used:

- Creation of applications

  Create applications logged in as an authorized user.

- Creation of a resource control program

  Create resource control programs logged in as an authorized user.

- Operation of the application

  Activate applications logged in as an authorized user.

Changing Access Permissions of Protected Resources

Change the access permissions of protected resources to prohibit access by users other than authorized users. To implement this measure, the aforementioned Construction of Environment by Specific Users must be done in advance. Perform this operation also logged in as an authorized user.

The following describes the procedure:

1) Stop the OTS system and resource control program.

   isstop -f

2) Change the access permissions for the protected resources.

The target protected resources are the following five:

- Folder storing the OTS system information
- Transaction log file
- Folder storing the trace log
- Repository storing the resource definitions
- RMP property file

Change the access permissions using [Property] of the protected resources. For the detailed procedure, refer to the manual of the OS.

Change the access permissions using the 'chmod' command. For the detailed procedure, refer to the manual of the OS.

3) Start the OTS system and resource control program.

   isstart

Periodic Backup

If you backup information periodically, you can restore the environment even if the information is tampered with. Periodic backup is an effective defense against the following threats:

- Tampering of information
- Damage to data
- Damage to file

There are two procedures for periodic backup:

- Data Backup
- Data Restoration

Data Backup

Use the 'otsbackupsys' command to perform periodic backups. By executing this command periodically, you can save information before damage is done on a resource to be protected. For more information about this command, refer to the Reference Manual (Command Edition).

Data Restoration

Restore the data when tampering or damage is detected in a resource to be protected. Use the 'otsrestoresys' command for restoration of data. Specify the desired file that is backed up, and restore it. For more information about this command, refer to the Reference Manual (Command Edition).

Use of the Security Function Provided by the Resource

When a password is transmitted in an environment open to the public like the Internet, the password may be read on the transmission path. You can minimize this risk by encrypting the password. The database linkage service guarantees consistency of transactions using the publicly available interface of each resource vendor.

- Decryption of passwords

For the details of the function, refer to the manual prepared by each resource vendor.

OLTP Function

The OLTP function can be used with the following products:

- Interstage Application Server Enterprise Edition
- Interstage Application Server Standard Edition
- Interstage Application Server Plus

This section gives an overview of the threats posed by invasion of security in a general OLTP application.

Generally, an OLTP application performs operations with a CORBA client program. This client program is executed sometimes as an independent CORBA client program and sometimes as an applet in a Web browser. Although it is usual to place the CORBA client program in an intranet area, a Web server ("HTTP Tunneling") acts as an intermediary to enable it to run if it is placed in an Internet area. This Web server is generally located in the Demilitarized Zone (DMZ) so that accesses to Internet and intranet areas go through a firewall.

Resources to be Protected

This section describes the resources to be protected when a general OLTP application is used.

Functions to be Protected

The following functions and procedures should be protected:

- User authentication
- Invocation of the CORBA application
- Invocation of the transaction application
- Access to Naming Service
- Access to Interface Repository
- Access to the load balance
- Interstage environment setup
- Registration and deletion of the WorkUnit definition

Resources to be Protected

The following table lists the resources when an OLTP application is used. If high security is required, it is desirable to protect these resources.

Table 1-9 Resources to be Protected

Function	Resource to be protected
User authentication	Password used for authentication
Invocation of the CORBA application	Log file for CORBA service Implementation Repository file Output file for the CORBA WorkUnit Log file for standard output Log file for standard error output
Invocation of the transaction application	Error log file WorkUnit definition Output file for the transaction application Standard output file for the file transaction application Standard error output file for the transaction application
Access to Naming Service	Data file for Naming Service
Access to Interface Repository	Data file for Interface Repository
Access to the load balance	Naming Service for load balance
Interstage environment setup	Definitions related to Interstage Interstage system definition file Interstage operating environment file CORBA Service operating environment file Component Transaction Service environment definition file
Registration and deletion of the WorkUnit definition	WorkUnit definition

Possible Threats to Resources

The following describes the possible security threats posed to resources to be protected in operation of an OLTP application.

Table 1-10 Possible Security Threats

Resource to be protected	Possible threat
User account used for authentication	Decryption of passwords Exploitation of passwords
Log file for CORBA service	Tampering of data recorded in the file Exploitation of information recorded in files Damage to files
Implementation Repository file	Tampering of data recorded in the file Exploitation of information recorded in files Damage to files
Output file for the CORBA WorkUnit	Tampering of data recorded Exploitation of information recorded Damage to data
Error log file	Tampering of data recorded in the file Exploitation of information recorded in files Damage to files
WorkUnit definition	Tampering of data recorded in the file Exploitation of information recorded in files Damage to files
Output file for the transaction application	Tampering of data recorded Exploitation of information recorded Damage to data
Data file for Naming Service	Tampering of data recorded in the file Exploitation of information recorded in files Damage to files
Data file for Interface Repository	Tampering of data recorded in the file Exploitation of information recorded in files Damage to files

Resource to be protected	Possible threat
Naming Service for load balance	Tampering of data recorded in the file Exploitation of information recorded in files Damage to files
Definitions related to Interstage	Tampering of data recorded in the file Exploitation of information recorded in files Damage to files
WorkUnit definition	Tampering of data recorded in the file Exploitation of information recorded in files Damage to files

Countermeasures Against Security Risks

The following describes possible countermeasures against security risks to the resources.

Table 1-11 Countermeasures

Possible threat	Countermeasures
Decryption of Passwords	Encryption of passwords
Exploitation of passwords	Encryption of passwords Periodic password change
Tampering of data recorded in the file	Setting access permissions on the file storing the information Periodic data backup
Exploitation of information recorded in files	Setting access permissions on the file storing the information
Damage to data	Periodic data backup
Damage to files	Setting access permission to the file

Countermeasures Against Decryption of Passwords

In an environment open to the public like the Internet, passwords may be decrypted on their transmission route. You can minimize this risk by encrypting passwords.

Countermeasures Against Exploitation of Passwords

In an environment open to the limited users like an intranet, it is not likely that passwords will be decrypted. Such an environment may be the management base of the passwords, and the information of passwords is often saved in a file. If this file is accessible by unauthorized users, there is a high risk of exploitation of the password information saved in the file. An effective countermeasure against this threat is to set appropriate access permissions on this type of file.

Countermeasures Against Tampering of Data Recorded in the File

There are environment definition files and other such files in the operating environment of an OLTP application. If the information in this file is illicitly tampered with, it may disable an OLTP application and cause various problems. An effective countermeasure against this threat is to set appropriate access permissions on this file. For Solaris OE system or Linux system, refer to Enhancing Security (Protecting Interstage Resources) in Appendix A.

Periodic backup is also effective. For the information about backup, refer to Maintenance (Resource Backup) in the Interstage Operator's Guide.

Countermeasures Against Exploitation of Information Recorded in Files

There are files storing information necessary for the operation of an OLTP application. The contents of these files are also a part of resources, and it is important to prevent exploitation of them. To minimize the risk of exploitation of information, it is effective to set appropriate access permissions on these files. For Solaris OE system or Linux system, refer to Enhancing Security (Protecting Interstage Resources) in Appendix A.

Countermeasures Against Damage to Data

In the operating environment of an OLTP application, there are important files like the environment definition file. If information in these files is illicitly tampered with, it may disable an OLTP application and cause various problems. An effective countermeasure against this threat is to set appropriate access permissions on these files. For Solaris OE system or Linux system, refer to Enhancing Security (Protecting Interstage Resources) in Appendix A.

Periodic backup is also effective. For backup, refer to Maintenance (Resource Backup) in the Interstage Operator's Guide.

Countermeasures Against Damage to Files

In the operating environment of an OLTP application, there are important files like the environment definition file. If information in these files is illicitly tampered with, it may disable an OLTP application and cause various problems. An effective countermeasure against this threat is to set appropriate access permission on these files.

Operation must be done by authorized users (Administrator) having administrator privileges. Change the permissions of the file to be accessible only to authorized users.

For the detailed procedure, refer to the manual of the OS.

Refer to Enhancing Security (Protecting Interstage Resources) in Appendix A, and take appropriate actions.

Periodic backup is also effective. For backup, refer to Maintenance (Resource Backup) in the Interstage Operator’s Guide.

Smart Repository

The Smart Repository function can be used with the following products:

- Interstage Application Server Enterprise Edition
- Interstage Application Server Standard Edition
- Interstage Application Server Plus

Resources Requiring Security Protection

This section explains the resources requiring security protection when Smart Repository is used.

Smart Repository Functions and Resources Requiring Protection

Smart Repository returns a result in response to an authentication request from a client.

Smart Repository has the following functions:

- User authentication function
- Password encryption function
- Entry search function
- Entry operating function
- Smart Repository operation function
- Logging function
- Operating environment definition

The following table indicates the resources requiring security protection for each function:

Function	Resources requiring protection
User authentication function	Authentication information (passwords) of the registered users (entries) Authentication information (password) for the Smart Repository administrator DN (identification information)
Password encryption function	Setting information for the user password encryption method
Entry search function	Passwords contained in search results
Entry operating function	Passwords contained in entries
Smart Repository operation function	Smart Repository
Logging function	Log related setup files Log data
Operating environment definition	Environment definition file

Potential Security Threats

The following indicates the potential security threats to the resources requiring Smart Repository protection:

Resources requiring protection	Potential threats
Authentication information (passwords) of the registered users (entries) Authentication information (password) for the Smart Repository administrator DN	Password decryption Password theft
Setting information for the user password encryption method	Illegal use of Smart Repository
Passwords contained in search results	Password decryption Password theft
Passwords contained in entries	Password alteration Password deletion
Log related setup files	Alteration of information contained in files File destruction
Log data	Alteration of information recorded in log data Log data destruction
Environment definition file	Alteration of information contained in files File destruction

Threats and Security Measures

In Smart Repository, the following measures can be taken to guard resources from security violation:

Threats	Security Measures
Password decryption Password theft	Password encryption Communication data encryption Periodic change of passwords
Illegal use of Smart Repository Password alteration Password deletion	Operation by limited users
Alteration of information contained in files Alteration of information recorded in log data	Periodic data backup
File destruction Log data destruction	Setting access rights for files

Password Encryption

When an entry search is requested from a client to Smart Repository, the password included in an entry can be retrieved in the form of an encrypted password string by using a method other than the original encryption method for user password encryption. Password encryption is a good way of protecting against the threat of password decryption.

Communication Data Encryption

When operation is requested from a client to Smart Repository, DNs, authentication information (passwords), and other communication data are used without being encrypted in the initial settings. The same applies to communication between a master and slave when the replication function is used.

SSL communication is used for the encryption of communication data on a communication path. By using SSL communication, SSL encryption can be a good measure for countering against the threats of password decryption and theft even if communication eavesdropping occurs.

For details on SSL communication, refer to "Method for Using SSL in Smart Repository."

Periodic Change of Passwords

It is possible that a password could be guessed or decoded by a malicious person (or computer) on the communication path. It is recommended that users observe the registration and operation rules for passwords used in user authentication.

A specific example of password registration rules:

- Use a password that is difficult to guess.
  - Use upper and lower case letters, special characters, and numeric characters together.
  - Avoid using personal information (names, nicknames, telephone numbers, date of birth, and so on).
  - Use eight or more characters for a password.
- Change passwords periodically. For example, change a password four times a year (every three months), and make sure that any new passwords are different from those used in the past.

Operation by Limited Users

As well as the threat of password decryption and theft, not remaining in place while logged in as the Smart Repository Administrator DN to the Entry Administration Tool can result in unauthorized operation.

An example of unauthorized operation:

- The password for an entry is altered or deleted.

To cope with such a threat, it is recommended that operation rules to limit users are established and observed by users.

A specific example of operation rules to limit users:

- The location in which the Entry Management Tool is used is a special location where access is controlled so that only permitted persons can enter and leave.
- When leaving their desks, users must log out or quit the Entry Administration Tool.
- When leaving their desks, users must enable the lockout function of their computer monitor screens.

Periodic Data Backup

By performing data backup periodically, the environment can be restored even if information is altered through unauthorized access. Periodic backup provides good protection against the following threats:

- Destruction or deletion of Smart Repository data
- Alteration of information recorded in files
- File destruction

Use the backup command (irepbacksys) to perform periodic backups. By periodically executing this command, information prior to the destruction of protection target resources can be saved, and a required generation can be restored even when resources are destroyed. For irepbacksys command details, refer to "irepbacksys" in the Reference Manual (Command Edition).

Setting Access Rights for Files

If program files, resource files, data or other files comprising Smart Repository are destroyed or deleted, services may stop or a program may be unable to start in Smart Repository. Setting appropriate access permission for files is an effective measure to protect against such threats. Avoid unnecessarily lowering the access permission levels that were set as initial settings.

Interstage Single Sign-on

This section explains the security threats for Interstage single sign-on and the countermeasures that can be taken.

Configuration Model

The figure below shows the basic configuration model for Interstage single sign-on.

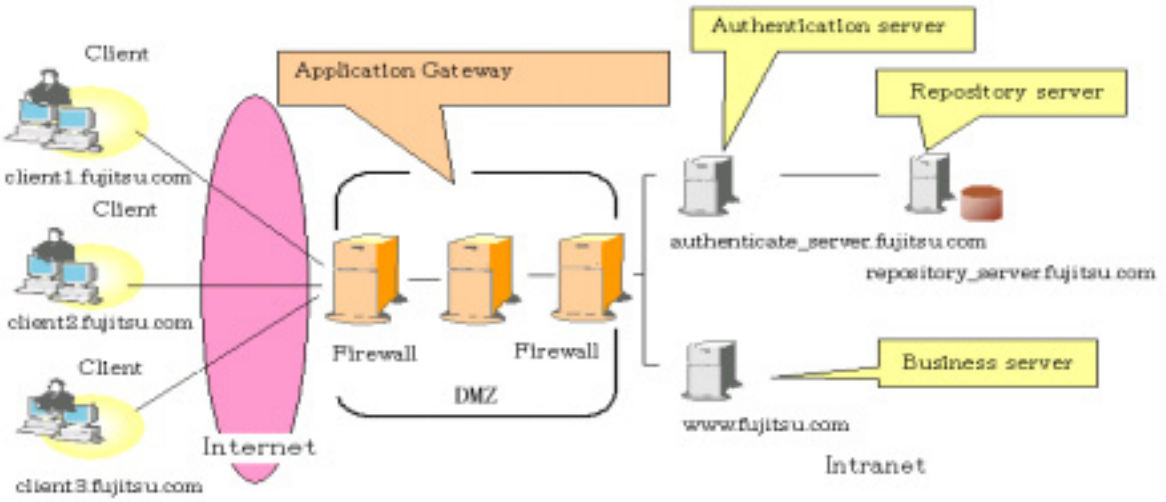


Figure 1-1 Interstage Single Sign-on System

The Interstage Single Sign-on system consists of three types of servers: repository servers, authentication servers, and business servers. The user uses the system from the client Web browser.

Each server executes Web server programs. Programs that perform single sign-on run on the Web server programs. The authentication and repository servers verify users, and the business server authorizes the use of protection resources and provides various business services. More than one business server can be established.

Configure the system so that clients accessing from the Internet go through the application gateway allocated in the demilitarized zone (DMZ). Allocate the servers in the intranet connection to protect the system from being accessed directly from the Internet.

Possible Threats

This section explains the possible threats when using Interstage Single Sign-on.

Deleting, Rewriting, and Exposing Server Resources

The repository server, authentication server, and business server contain important files to control the programs. The files include the authentication infrastructure setup file and business system setup file required for setting up each server, and the configuration file and service ID file created after setting up the servers. The possible threats to these resources are as follows:

- Deletion of resources, which disables system configuration and operation.
- Rewriting of resources, which causes results not intended by the administrator (e.g., disabled authentication or authorization).
- Exposure of resources, which causes user spoofing or system takeover.

Rewriting and Exposure of Communication Contents

Important data items (e.g., user name, password, and authentication or authorization control information) are exchanged between the servers or between a Web browser (client) and a server. If these data items are rewritten, authentication or authorization may be controlled incorrectly. Interception of such data involves the risk of password leakage or spoofing.

Communication contents could be leaked by network interceptors, or by someone tapping into the information on the proxy server log or business server access log along the communication route.

User Spoofing

Interstage Single Sign-on verifies users with one or both of two authentication methods: certificate authentication and password authentication using the user name and password.

Certificate authentication requires a security key paired with the certificate. Leakage of this key may cause spoofing. Similarly, password authentication requires the user name and password, leakage of which may cause spoofing. It is particularly dangerous to use a simple password because it can be guessed and tried by others relatively easily. The attacker may use a special program to make a dictionary or use the brute force attack method to decode.

Authentication Server Spoofing

In password authentication, the authentication server asks the user to provide the user name and password. In practical terms, the Web browser prompts the user to enter the user name and password in the dialog box. The authentication request is usually issued when the user accesses the business server via a Web browser.

However, it is difficult to confirm that the Web browser requesting the user name and password is representing the proper authentication server. It is possible that an attacker posing as the authentication server could trick users into entering their names and passwords. Users could unknowingly give out their user names and passwords to a server prepared by the attacker.

DoS Attack

In a denial of services (DoS) attack, the attacker generates a large amount of accesses to the system to create heavier loads for the server. This leads to slower response resulting in deterioration of the service quality, or excludes regular users from using the services.

Application Risk

Interstage Single Sign-on stores important information in the Web browser cookie. The attacker could collect cookies for spoofing when the application operating on the business server is vulnerable, e.g., cross site scripting (XSS) or allocation of a malevolent application.

Client Risk

When an attacker takes advantage of Web browser defects and obtains cookie information, vulnerability may become apparent. This could pose a threat to Interstage Single Sign-on because the user uses a Web browser for the client.

Information Leakage Threat

Interstage Single Sign-on allows users to customize the messages displayed on the Web browser according to the user environment. The information in this message could provide an opportunity for attackers.

Security Measures

This section explains the action require to handle assumed threats.

Protecting the Authentication Infrastructure Setup File and Business System Setup File

The authentication infrastructure setup file and business system setup file are required for setting up a repository server, authentication server, and business server. Manage the files (and the password specified when downloading them) so that they cannot be leaked to a third party, and transfer them by safe means.

The downloaded authentication infrastructure setup file and business system setup file are encrypted using a password specified when downloading them. Exposure of the file contents may cause user spoofing or system takeover, so always delete these files once the server configuration is complete.

Setting Access Permission for Operating Resources

To protect operating resources on servers, appropriate access permissions must be established for the operating resources. Minimize the number of users or programs that can access the resources to protect them from deletion, rewriting, or exposure by an attacker.

Interstage Single Sign-on grants appropriate access permissions to operating resources. When changing the effective user of the Web server, also change the access permissions.

The administrator may delete operating resources by mistake, so periodically back up the operating resources.

To change the effective user of the Web server, see "Operation and Maintenance" in the Single Sign-on Operator's Guide. To back up operating resources, see the Operator's Guide.

Protecting Communication Contents

Encryption is an effective way of protecting communication contents from being rewritten or exposed. Use https as the protocol for the authentication and business servers and encrypt the communication contents. The SSL environment is required to operate https. The repository server need not use SSL communication because the Interstage Single Sign-on program encrypts the communication contents.

To set up environments for the authentication and business servers using https, see "Environment Setup (SSO Administrators)" and "Environment Setup (Business Server Administrators)" in the Single Sign-on Operator's Guide.

To operate the system more securely and prevent DoS attacks, install IPSec communication or a firewall to protect the authentication and repository servers. For details, see "More Secure Use" in the Single Sign-on Operator's Guide.

Confirming the Authentication Server

It is necessary to advise users not to enter user names or passwords, by mistake, into the form authentication page or password authentication dialog represented by a false authentication server.

- Announce the correct authentication server host name (URL) to users.
- Have users access the business server via bookmarks set in their Web browsers or from links on reliable Web sites.
- When the requesting source in the basic authentication dialog could be faulty, make users cancel the dialog and confirm whether the URL displayed in the address display area of the Web browser matches that of the correct authentication server.
- If the requesting source of the form authentication page displayed by the Web browser is not clear, have the user confirm that the URL displayed in the address display area of the Web browser matches that of the correct authentication server.

Countermeasures Against Password Attacks

The password used for authentication may be stolen and abused. Password theft includes brute force attack and dictionary attacks using hacking tools. To operate the system more securely, educate users using the items described below.

In a practical sense, add the operation requirements and security policy of the target system to determine the rules.

Interstage Single Sign-on consolidates the management of resource information using the Smart Repository. The administrator DN authentication information (password) required for using the Smart Repository must also be protected. See "Smart Repository" for details of the threats and countermeasures for the authentication information (password) of the Smart Repository administrator DN.

Difficult-to-guess Password

Use a password that cannot be easily guessed by others or identified mechanically by some kind of tool.

A difficult-to-guess password should meet the following conditions:

- Cannot be guessed from personal information, e.g., name or birthday.
- Comprises the longest character string possible.
- Contains uppercase and lowercase letters, numbers, and symbols.
- Contains one complete word without modification.
- Is not a simple character string such as a repetition of the same character.

Password Management Method

The password must not be known to others. The following actions are very unwise:

- Disclosing the password to others.
- Leaving a note containing the password where others can see it.
- Storing the password in the Web browser.

Periodical Change of the Password

Even if the above two items are addressed, the password may be leaked. Periodically change the password to ensure secure operation.

Note

Interstage Single Sign-on does not provide a password change function. Educate users to change the password with appropriate frequency according to the system to be used.

If a password is guessed or stolen by others, stop all business servers and confirm whether the re-authentication interval that was set with either of the following operations has passed:

- Re-authentication interval for each user, set in ssoCredentialTTL of user information in the SSO repository.
- Standard re-authentication interval, configured in the Interstage Management Console, [Security] > [Single Sign-on] > [Authentication infrastructure] > [Authentication server] > [Settings] tab > [Detailed Settings [Show]] > [Operation after Authentication] > [Re-authentication Interval].

After the re-authentication interval has passed, change the guessed or stolen password and restart all business servers.

Operating and Managing a Business Server

To prevent unauthorized access to the protection resources of the business server and control correct authentication or authorization, the business server must be operated and managed appropriately.

- Use https as the protocol of the business server and encrypt the communication contents with the Web browser.
- Do not operate an unnecessary Web service on another port of the same server.
- Create a service ID of the business server using FQDN of the business server.
- Before deploying an application such as CGI on the business server, verify that there is no security problem such as XSS.
- Minimize the number of users that can log in to the business server and record login actions.

Application Programming

Confirm that the application to be operated on the business server does not show vulnerability (such as buffer overflow or XSS) and is securely programmed.

This measure applies to web applications in general, and is not limited to Interstage Single Sign-on.

Interstage provides single sign-on JavaAPIs. With single sign-on JavaAPIs, an application that performs authentication can be created based on the user name and password, and the user information stored in the SSO repository. Because such an application uses important information such as user names and passwords, be aware of the risk of information leakage caused by application failures. Use care when developing and operating applications.

When using single sign-on JavaAPIs, the following threats are assumed. Take the appropriate action for each of these threats:

For Servlet Applications Using Single Sign-on JavaAPIs

Possible threat	Action
Application alteration	- Periodically change the password for the IJServer operation account to prevent it from being leaked or collected.
Application destruction	- Periodically back up data.
Leakage of credential information, user IDs, and passwords	- Use the Web server in SSL. - Minimize the number of access permissions to Web server access log files. - Use POST for the FORM method instead of GET
Alteration or exposure of configuration files (security policy file, login configuration file, or trust store file)	- Minimize the number of access permissions to operating resources.
Destruction of configuration files (security policy file, login configuration file, or trust store file)	- Periodically back up data.

For Java Applications Using Single Sign-on JavaAPIs

Possible threat	Action
Application alteration	- Periodically change the account password.
Application destruction	- Periodically back up data.
Leakage of user IDs or passwords	- Securely implement communication with the client (using SSLSocket, etc.)
Operation of the application operating terminal	- When operating Java applications in server applications, implement them as daemon processes or services.
Alteration or exposure of configuration files (security policy file, service ID file, login configuration file, or trust store file)	- Minimize the number of access permissions to operating resources.
Destruction of configuration files (security policy file, service ID file, login configuration file, or trust store file)	- Periodically back up data.

Applying Patches

Periodically check failure information regarding Web browsers and operating systems. If a new failure is detected, patches or workarounds are made available. Remind users to apply the latest security patches to their Web browsers. Similarly, apply the latest fix to the operating system of each server.

Messages Displayed on the Web Browser

When customizing messages to be displayed on users' Web browsers, use particular caution in regard to the contents.

- Avoid including information that could be used as a stepping stone or clue to attacks.
- Before displaying telephone numbers, mail addresses, or URLs, verify that there will be no problem in publicizing such information.

Multi Server Management

This section describes how to deal with security threats using Multi Server Management.

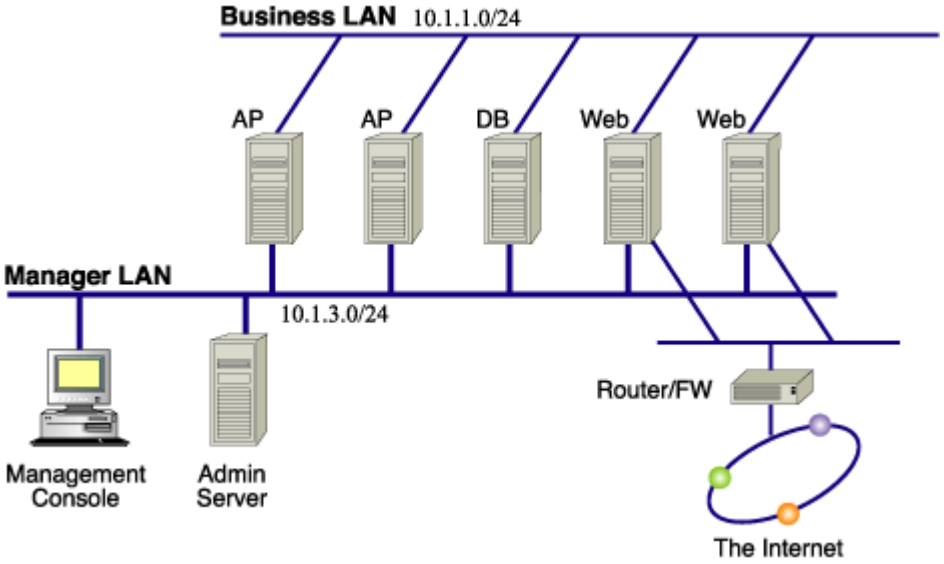
The Admin Server function of Multi Server Management can only be used with the following products:

- Interstage Application Server Enterprise Edition
- Interstage Application Server Standard Edition
- Interstage Application Server Plus

The Managed Server function of Multi Server Management can be used with all products.

Configuration Model

When using Multi Server Management, the LAN for the flow of the actual business data and the LAN for the flow of operation management data between the Admin Server and Managed Server are usually separated.. The former is called the “business LAN”, and the latter the “management LAN”. The following figure shows an overview of the business LAN and management LAN and a typical usage model for Multi Server Management.



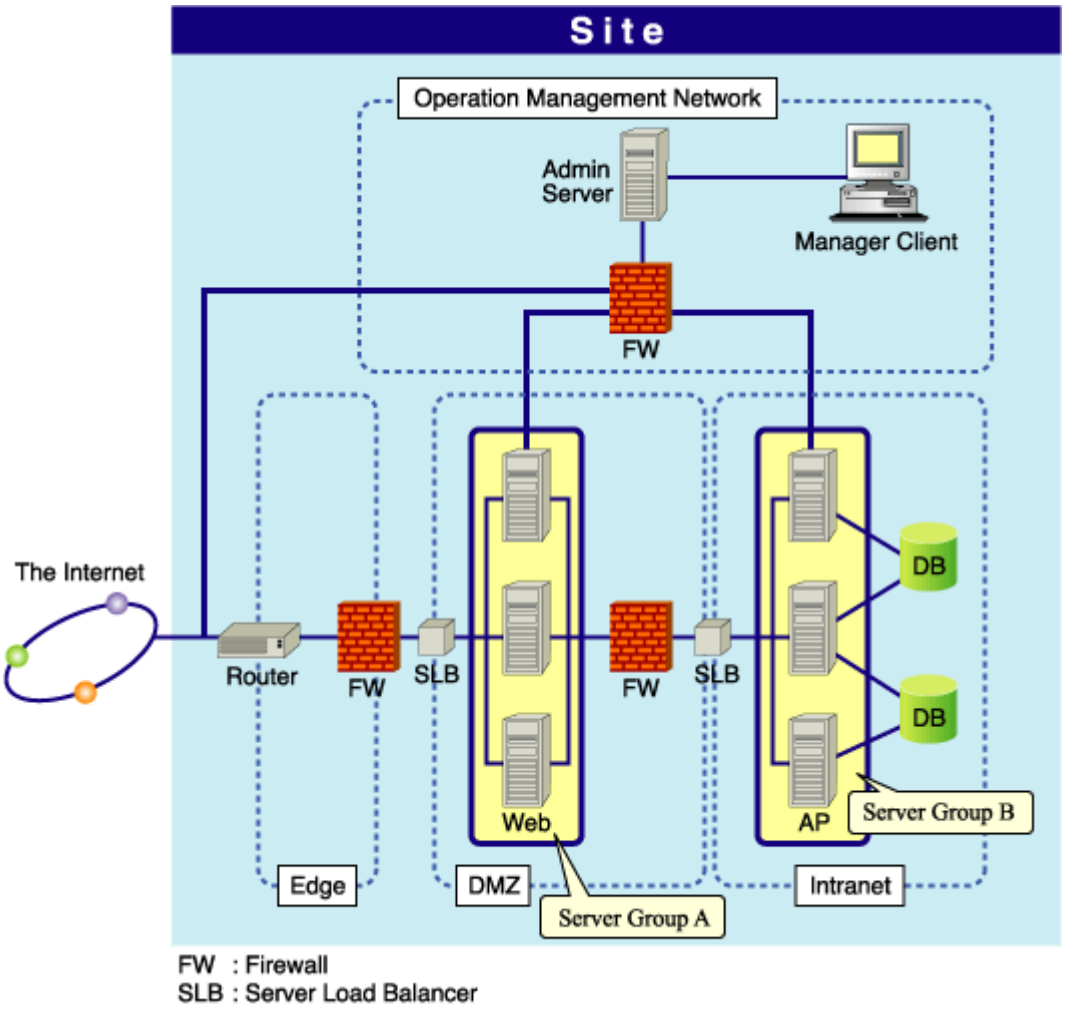


Figure 1-2 Multi Server Management Configuration Model

In a typical Multi Server Management configuration, one Admin Server manages one site. The site is configured using multiple servers, with servers that execute the same business applications grouped together. The Admin Server runs the servers in the site using the Interstage Management Console.

Apart from the fact that there is an Admin Server managing site operation (and that the business application is configured using multiple servers), there is no difference from the normal Interstage usage model. Refer to the operation model and Security Measures chapter for each Interstage function.

Resources to be Protected

This section describes the resources to be protected when Multi Server Management is used.

Functions to be Protected

The following functions and procedures should be protected:

- Operations using the Interstage Management Console

Resources to be Protected

The following resources should be secured to protect the functions they provide:

Table 1-12 Resources to be Protected

Function	Resource
User authentication	User ID and password used for authentication
Connection to Web server (When Interstage HTTP Server is used)	Definition file for Interstage HTTP Server
Connection to Web server (When InfoProvider Pro is used)	Definition file for InfoProvider Pro
Interstage Operation Tool environment setup	Definition file for the Interstage Operation Tool

Possible Security Risks to Resources

The following security threats may occur during operation of the Interstage Management Console and Interstage Operation Tool.

Table 1-13 Possible Security Risks to Resources

Resource	Possible threat
User authentication	- Exploitation of user IDs and passwords - Decryption of user IDs and passwords
Definition file for Interstage HTTP Server	- Tampering with data recorded in the file - Exploitation of information recorded in files - Damage to files
Definition file for the Interstage Operation Tool	- Tampering with data recorded in the file - Exploitation of information recorded in files - Damage to files

Threat Prevention

The following table lists countermeasures that can be taken against possible security risks.

Table 1-14 Threat Countermeasures

Possible threat	Countermeasures
Decryption of user IDs and passwords	- User ID and password protection
Exploitation of user IDs and passwords	- Setting an expiration date for the user ID and password
Tampering of data recorded in the file	- Setting access permissions for the file storing the information - Periodic data backup
Exploitation of information recorded in files	- Setting access permissions for the file storing the information

Countermeasures Against Decryption of User IDs and Passwords

In an environment open to the public like the Internet, user IDs or passwords may be decrypted on their transmission route. The Interstage Management Console and Interstage Operation Tool implement encryption of user IDs and passwords, but it is still possible for them to be decrypted. To minimize this risk, set expiration dates for user IDs and passwords and change them periodically.

Countermeasures Against Exploitation of User IDs and Passwords

In an environment open to limited users like an intranet, it is not likely that user IDs and passwords will be decrypted. Such an environment is often the management base of user ID and password information, and user ID and password information is often saved in a file. If this file is accessible by unauthorized users, there is a high risk of exploitation of user ID and password information. An effective countermeasure against this threat is to set appropriate access permissions for files, storing user ID and password information.

Countermeasures Against Tampering of Data Recorded In Files

To use the Interstage Management Console and Interstage Operation Tool, the Interstage HTTP Server environment definition file is required. If the information in this file is tampered with, it may disable the Interstage Management Console and Interstage Operation Tool and cause various problems. An effective countermeasure against this threat is to set appropriate access permissions for this file. For Solaris OE or Linux systems, refer to Enhancing Security (Protecting Interstage Resources) in Appendix A.

Periodic backups are also effective. For backup information, refer to Maintenance (Resource Backup) in the Interstage Operator's Guide.

Countermeasures Against Exploitation of Information Recorded in Files

The information required for operation of the Interstage Management Console and Interstage Operation Tool is stored in files. The contents of these files are also resources, and it is important to prevent exploitation of them. An effective means of protecting these files is to set appropriate access permissions for them. For Solaris OE or Linux systems, refer to Enhancing Security (Protecting Interstage Resources) in Appendix A.

Countermeasures Against Damage to Files

There are important files, such as the environment definition file, in the Interstage Management Console and Interstage Operation Tool environment. If information in these files is tampered with, it may disable the Interstage Operation Tool and cause various problems. An effective countermeasure against this threat is to set appropriate access permissions for these files. For Solaris OE or Linux systems, refer to Enhancing Security (Protecting Interstage Resources) in Appendix A.

Periodic backups are also effective. For backup information, refer to Maintenance (Resource Backup) in the Operator's Guide.

Configuration Management Function

This section describes how to deal with security threats using the Configuration Management function.

Configuration Management Function Usage Model

The following figure shows a typical usage model for the Configuration Management function.

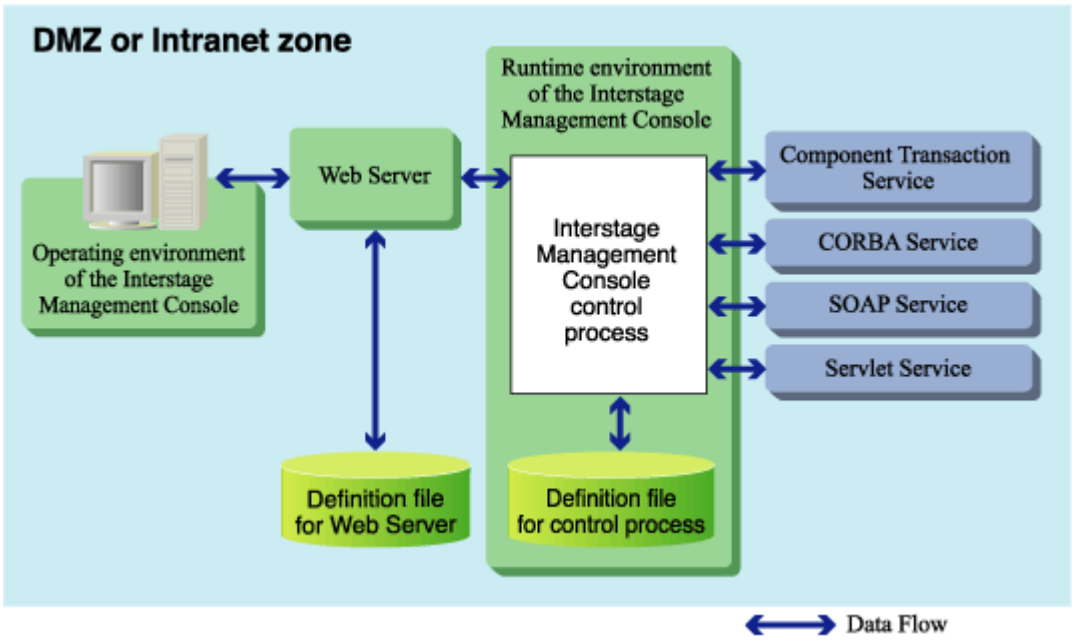


Figure 1-3 Configuration Management Function Usage Model

The Configuration Management function stores operations in the Interstage Management Console internally. For this reason, users do not need to be aware of the configuration of the configuration management function. This chapter explains the settings information for the Environment Settings window in a repository that is available to the system administrator, and also explains security breach threats.

Resources to be Protected

This section explains the resources that should be protected for the repository that is managed by the Configuration Management function.

Functions to be Protected

The following functions and procedures should be protected:

- Operations using the Interstage Management Console

Resources to be Protected

The following resources are used in the Interstage Management Console. If advanced security measures are requested, it is advisable to protect these resources as part of that security.

Table 1-15 Resources to be Protected

Function	Resources that should be protected
Various operations using the Interstage Management Console	- Information stored in the repository

Possible Security Risks to Resources

Additional possible security threats to resources that should be protected against using the Configuration Management function are shown below.

Table 1-16 Possible Threats to Resources

Resources that should be protected	Additional possible threats
Information stored in the repository	- Overwriting the information that is recorded in files - Exploiting the information that is recorded in files - File corruption

Threat Prevention

The following table lists countermeasures that can be taken to prevent against security breaches using the Configuration Management function.

Table 1-17 Threat Countermeasures

Possible threat	Countermeasures
Overwriting the information that is recorded in files	- Implement access authority settings for the files in which information is saved - Perform regular data backups
Exploiting the information that is recorded in files	- Implement access authority settings for the files in which information is saved
File corruption	- Implement access authority settings for the files

Countermeasures Against Overwriting Information Recorded in Files

Various items of Interstage information are saved in the Configuration Management function repository in binary format. If the contents of these files are overwritten illegally, it might cause various problems, such as being unable to run Interstage. To effectively counter this type of threat, implement appropriate access authority settings for the files in which this information is saved. Regular data backups are also an effective preventative measure against files being overwritten illegally.

Countermeasures Against Exploiting Information Recorded in Files

Various items of Interstage information are saved in the Configuration Management function repository in binary format. The descriptive content of these files is also a part of the resources, and it is important to prevent exploitation. To effectively counter this type of threat, implement appropriate access authority settings for the files in which this information is saved.

Countermeasures Against File Corruption

In the Configuration Management function repository, there are files that determine the validity of information based on whether or not a file can be read. If these files are corrupted for some reason, registered information cannot be referenced. To effectively counter this type of threat, implement appropriate access authority settings for the files.

Chapter 2

Security Measures

Generally, the services alone cannot completely protect resources from security attacks. Taking operational measures can also increase safety.

This chapter explains the security measures indicated in "Security Risks and Measures" separately for each service. To implement safe and firm operation against security violation, it is recommended to refer to and carry out the measures for the services used.

Security information on Fujitsu products is available from the following site. Keep checking the latest information.

http://software.fujitsu.com/en/security/main.html

Common Security Measures

This section explains the following topics:

- Notes on User Accounts
- Backup
- Notes on Interstage Installation Resources

Notes on User Accounts

To prevent termination of operation, alteration of resources, and leakage of information that may be done by end users, it is recommended not to register a user account that is not an authorized Administrator.

Backup

Periodic backup is recommended to minimize damage caused by external attacks, machine trouble, operation errors, and so on.

Notes on Interstage Installation Resources

To prevent alteration of resources by end users, it is recommended not to set access permissions like "Everyone (full control)" that permits access from unspecified users to the folders and files under the Interstage installation.

The "issetfoldersecurity" command changes the permissions of the Interstage installation folder and subfolder. For details, refer to the Reference Manual (Command Edition).

Security Measures for Interstage Operation Tool

The Interstage Operation Tool can be used with the following products:

- Interstage Application Server Enterprise Edition
- Interstage Application Server Standard Edition
- Interstage Application Server Plus

Notes on User Accounts

Operation of the Interstage Operation Tool by end users is restricted by giving permissions only to the users within the Administrators group to operate important functions such as activation and termination of Interstage. In addition to this, it is recommended to change passwords periodically to cope with possible problems like leakage of the Administrators group account information.

Notes on the Permissions of the Environment Definition File

To prevent alteration of resources and leakage of information by end users, it is recommended to set access permissions accordingly.

Notes on Communication Data

It is recommended to use SSL encryption for server-client data. .

Security Measures for Operation of the Web Server (Interstage HTTP Server)

This section explains the following topics:

- Notes When Making Access
- Notes on Communication Data
- Threats of Denial of Service Attacks (DoS)
- Leakage of Password Information
- Unauthorized Access to Resource Files
- Risk of Exploiting the HTTP TRACE Method
- Threat that the UNIX account name will be discovered

Notes When Making Access

When an access is made from a Web browser to the Interstage HTTP server, there is a possible threat that an ill-intentioned person could make an unauthorized access to the Interstage HTTP Server by impersonating a user having proper access permission.

To prevent this, SSL encryption using SSL version 3 (client authentication) is recommended.

For information about SSL encryption, refer to Chapter 9, How to Use SSL with Interstage HTTP Server.

Notes on Communication Data

An ill-intentioned person could access communication data between the server and a user who has proper access permission.

SSL encryption is recommended in order to minimize this type of risk.

For information about SSL encryption, refer to Chapter 9, How to Use SSL with Interstage HTTP Server.

Threats of Denial of Service Attacks (DoS)

An ill-intentioned person on the network could target a server and disable its services. To defend the server from Denial of Service attacks (DOS), it is recommended to use the following functions:

- User authentication:

  For information about user authentication, refer to User Authentication in Authentication and Access Control for the Interstage HTTP Server in Chapter 9.

- IP access control:

  It is possible to permit access only to specific clients.

  For information about IP access control, refer to IP Access Control in Authentication and Access Control for the Interstage HTTP Server in Chapter 9.

- Use of SSL encryption:

  High level of security can be retained, where client authentication is possible.

  For information about SSL encryption, refer to Chapter 9, How to Use SSL with Interstage HTTP Server.

- Limitations on the size of request message from client:

  Set the maximum size of a request message to prevent a buffer overflow. The maximum size of the request message is set by the following directives of the environmental definition file (httpd.conf):

  - LimitRequestBody
  - LimitRequestFields
  - LimitRequestFieldsize
  - LimitRequestLine

Leakage of Password Information

The Interstage HTTP Server has a password file, which an ill-intentioned person may furtively look into.

The password data in the password file is encrypted; still, it is recommended that the administrator create the password file using the 'htpasswd' command to make it inaccessible by end users.

Unauthorized Access to Resource Files

Interstage HTTP Server has resource files listed below:

- Contents
- Environment definition file (httpd.conf)
- Access log file
- Error log file
- CGI
- Environment definition file for each directory (.htaccess)

These files may be exposed to the threat of unauthorized access.

To protect these files, make these files inaccessible by end users. Making this file accessible only to users with administrator privileges is recommended (superuser for a Solaris OE/Linux system, and Administrator for Windows(R) system).

Risk of Exploiting the HTTP TRACE Method

Malicious users (or machines) on the network may read private information in HTTP request data or execute unwanted codes.

To prevent this risk, it is recommended to disable the HTTP TRACE method by specifying the following lines in the Interstage HTTP Server environment definition file (httpd.conf):

The TRACE method is the HTTP/1.1 method of receiving the data sent from the client side as response data. This method is used to diagnose the network environment. There is no problem in the Interstage HTTP Server operation if this method is disabled because it not usually used.

Windows

```
LoadModule rewrite_module modules/mod_rewrite.so
AddModule mod_rewrite.c
RewriteEngine On
RewriteCond %{REQUEST_METHOD} ^TRACE
RewriteRule .* - [F]
```

Solaris OE Linux

```
LoadModule rewrite_module libexec/mod_rewrite.so
AddModule mod_rewrite.c
RewriteEngine On
RewriteCond %{REQUEST_METHOD} ^TRACE
RewriteRule .* - [F]
```

Add the setting to each virtual host as follows to disable the HTTP TRACE method when virtual hosts are configured.

Windows

```
LoadModule rewrite_module modules/mod_rewrite.so
AddModule mod_rewrite.c
<VirtualHost 192.168.0.2>
    ServerName virt.example.com
    ServerAdmin webmaster@virt.example.com
    RewriteEngine On
    RewriteCond %{REQUEST_METHOD} ^TRACE
    RewriteRule .* - [F]
    ...
</VirtualHost>
```

Solaris OE Linux

```
LoadModule rewrite_module libexec/mod_rewrite.so
AddModule mod_rewrite.c
<VirtualHost 192.168.0.2>
    ServerName virt.example.com
    ServerAdmin webmaster@virt.example.com
    RewriteEngine On
    RewriteCond %{REQUEST_METHOD} ^TRACE
    RewriteRule .* - [F]
    ...
</VirtualHost>
```

Threat that the UNIX account name will be discovered

Solaris OE Linux

There is always a risk that a UNIX account name on the Web server will be discovered on the network by a user (or machine) with malicious intent.

To counter this kind of threat, it is recommended that the settings in the Interstage HTTP Server environment settings file (httpd.conf) are made as shown below. These settings will disable requests to documents under the UNIX account user home directory.

```
LoadModule  userdir_module  libexec/mod_userdir.so
AddModule   mod_userdir.c
UserDir     disabled
```

Note

A hash mark (#) can be added to the start of the LoadModule and AddModule directives to make the line a comment.

To make a document under the UNIX account user home directory public, configure the following settings:

- Set the access authority in the home directory to be made public for access from the Web server.
- Disable the user directory settings for users that are not going to be made public.

An example of disabling the user directory settings for users that are not going to be made public is shown below.

Example:

Making “user1” and “user2” documents under “user home directory/public_html” public.

```
UserDir public_html
UserDir disabled
UserDir enabled user1 user2
```

Making all documents, except for “user3” and “user4”, under “user home directory/public_html” public.

```
UserDir public_html
UserDir disabled user3 user4
```

Notes:

If just “UserDir public_html” is specified, when the "http://host name[:port number]/~user" request is received, the status code that is returned when the user name is specified as “user” depends on whether the user exists in the UNIX server. For this reason, the UNIX account name on the Web server might be discovered. These status codes are shown below.

- “user” does not exist:

  The “404 Not Found” status code is returned.

- “user” exists:

  The “403 Forbidden” status code is returned.

  This status code is returned because, although “user” exists, access authority for access from the Web server for this user has not been set in the home directory. Specify users that can execute the Web server in the User directive.

Linux

This problem occurs when the user home directory is created using the useradd command, and directory authority is only set for the owner, meaning that only that user has access permission.

Security Measures for Operation of the Web Server (InfoProvider Pro)

Windows Solaris OE

The InfoProvider Pro can be used on the Windows(R) system or Solaris OE system.

Notes on Permissions of Contents

Solaris OE

To prevent alteration by end users, it is recommended to set the proper permissions on the top-level directory to be made accessible and the directory that stores applications.

Notes on the Permissions of the Environment Definition File

Solaris OE

To prevent alteration of resources and leakage of information by end users, it is recommended to set the same permission as the sample environment definition to a newly created environment definition file.

Notes on User Authentication

Windows Solaris OE

The information of user IDs and passwords used for user authentication is transmitted without encoding. To prevent interception of user IDs and passwords, SSL encryption is recommended.

In addition to this, it is recommended to set rules for setting passwords and to make sure users follow this rule.

Example:

- Establish the password setting rule as follows, and set a password that the others cannot guess.
  - Mix upper-case and lower-cases letters, special characters, and numerals.
  - Do not use personal information (such as name, nickname, telephone number, birthday, etc).
  - Use eight or more characters.
- Change your password periodically. For example, change your password four times a year (every three months).

Security Measures for the Servlet Service

This section explains the following topics:

- Notes on the Use of Sessions
- Notes on Web Application Development
- Notes on Deployment of Web Applications
- Notes on the Root Directory of the Web Application
- Notes on Communication Data

Notes on the Use of Sessions

Session information is embedded in cookies or URL parameters.

When the Web server is connected to a Web browser via the Internet, the contents of communication are in danger of interception or alteration.

Therefore, SSL encryption is recommended.

Notes on Web Application Development

For notes on web application development, refer to "Common Notes for Interstage" in the Product Notes.

Notes on Deployment of Web Applications

It is recommended to give write permissions only to users who execute the Servlet container to prevent alteration by end users.

Notes on the Root Directory of the Web Application

If a directory open to the public on the Web server is the same as the root directory of the Web application, the body of the Web application including the class files and Jar files may be accessible through the Web browser.

To prevent this problem, it is recommended to make the directory made open by the Web server different from the root directory of the Web application.

Notes on Communication Data

Possible threats to communication between the Web server connector and Servlet container are as follows:

- The Web server connector is impersonated to illegally access the Servlet container.
- Communication data is viewed by unauthorized person.
- Communication data is altered.

It is recommended to use SSL communication for protection from these threats. SSL version 3 (client authentication) is required for protection from impersonation.

Refer to "Environment setup for Servlet service" for information on enabling SSL communication.

Security Measures for the EJB Service

This section gives an outline of security risks when the EJB service is used.

EJB Service is required when the "EJB function" of J2EE applications is used.

Web-J supports only the client function of the EJB service.

Resources to be Protected

This section describes the resources to be protected when the EJB service is used.

Resources to be Protected

The table below lists the resources that are used for EJB Service. When high security is required, protect these resources for security.

Table 2-1 Resources to be Protected

Function	Resource to be protected
EJB environment setup	Environment definition file of EJB Service
EJB application program operation	Environment definition file of EJB Service J2EE common directory

The following describes the locations of these resources. The directory structure of a Windows(R) system is taken for example.

- Environment definition file of EJB Service

  All files under C:\Interstage\EJB\etc directory

  All files under C:\Interstage\J2EE\etc directory
- J2EE common directory

  All files under J2EE common directory

Solaris OE /Linux system is taken for example.

- Environment definition file of EJB Service

  All files under /opt/FJSVejb/etc directory

  All files under /opt/FJSVj2ee/etc directory
- J2EE common directory

  All files under J2EE common directory

Possible Threats to Resources

The following countermeasures can defend EJB Service against security invasion.

Table 2-2 Possible Threats to Resources

Resource to be protected	Threats
Environment definition file of EJB Service	Tampering of information Exploitation of information Damage to data Damage to files
Application folder	Tampering of information Exploitation of information Damage to data Damage to files Exploitation of passwords

Countermeasures Against Threats

The following countermeasures can be used to minimize security risks for the EJB Service.

- Operation confined to authorized users
- Periodic backup
- Use of SSL encryption

Confining Operation to Specific Users

Confining operations to a limited set of users can be an effective defense against following threats:

- Tampering of information
- Exploitation of information
- Damage to data
- Damage to files
- Exploitation of passwords

Operation confined to specific users implements the following two procedures:

- Selection of the users
- Change of access permission to the protected resources

Selection of Specific Users

By fixing the operators of the entire system to a pre-specified set of users, you can prevent tampering of information.

Executing the security enhancement command, you can change the operation mode of EJB Service to the specific user authorization mode.

For detailed information about the security enhancement command, refer to Environment Setup for Interstage Resources Protection in Appendix A.

Change of Access Permission of the Protected Resources

Change the access permission of the resources to be protected.

Periodic Backup

Periodic backups make restoration of the environment possible when information is tampered with. Periodic backup is an effective defense against following threats:

- Tampering of information
- Damage to data
- Damage to files

Periodic backup implements the following two procedures:

- Backup of data
- Restoration of data

Backup of Data

In preparation for tampering, back up the following resources periodically:

- Environment definition file of EJB Service
- Application folder

For detailed information about backup of the resources of EJB Service, refer to Maintenance (Resource Backup) in the Interstage Operator's Guide.

Restoration of Data

If data has been damaged, restore the data.

SSL Encryption

The SSL encryption function is a function for data encryption using the SSL linkage of CORBA Service. Using the SSL linkage function, you can defend your system against the following threats:

- Exploitation of information
- Exploitation of passwords

For information about SSL encryption, refer to Chapter 11, How to Use SSL with J2EE.

Security Measures for J2EE Deployment Tool

This topic explains the following topic:

- Unauthorized Access to Resource Files

Unauthorized Access to Resource Files

The J2EE Deployment tool is used for deploying J2EE applications and making them usable. The J2EE applications to be deployed are saved in ear file, jar file, rar file, and war files. These files may be exposed to the threat of unauthorized access from an ill-intentioned person.

To protect these files from this threat, make these files inaccessible by end users. For this purpose, it is recommended to allow access only by users having administrator authorization (superuser for a Solaris OE/Linux system, and Administrator for Windows(R) system).

Security Measures for the J2EE Resource Access Definition

This section explains the following topic:

- Leakage of Password Information

Leakage of Password Information

The J2EE resource access definition can hold definitions of access information for various resources used by J2EE applications. This access definition information is saved in a file, which includes password information. There is a possible threat that an ill-intentioned person may furtively read this file.

A countermeasure for defending the file storing password information from threats is to make it inaccessible by end users. For this purpose, it is recommended to set a rule that only users having administrator authorization (superuser for a Solaris OE/Linux system, and Administrator for Windows(R) system) can use the J2EE resource access definition.

Security Measures for Interstage JMS

Interstage JMS can be used with the following products:

- Interstage Application Server Enterprise Edition
- Interstage Application Server Plus

Unauthorized Access to Resource Files

Interstage JMS Server has environment definition files as listed below:

- JNDI definition file (fjmsjndi.ser.*) (*1)
- JMS non-volatilization file (fjmsmng.ser.*, fjmsdsubXXXX.ser, lock\.XXXX) (*1)
- Cluster environment definition file (fjmscluster.ser) (*1)
- Console log (fjmsconsole.log)

*1 For the locations where the files are stored, refer to "Backing Up and Restoring Resources" - "Outline and Applicable Resources" in Maintenance (Resource Backup) in the Operator's Guide.

These files may be exposed to the threat of unauthorized access from an ill-intentioned person.

To protect these files from this threat, make these files inaccessible by end users. For this purpose, it is recommended to allow access only users having administrator authorization (superuser for a Solaris OE/Linux system, and Administrator for Windows(R) system).

Security measures must also be taken for Event Service. For information about Event Service, refer to Security Measures for Event Service.

Security Measures for CORBA Service

This section explains the following topics:

- Unauthorized Access to Resource Files
- Notes on Communication Data
- Notes on the Port Number used by CORBA Service
- Notes on Creation and Operation of Java Applets

Unauthorized Access to Resource Files

CORBA service has environment definition files as listed below:

- CORBA Service
  - CORBA Service environment definition information file (config) (*1)
  - Host information definition file (inithost/initial_hosts) (*1)
  - Server default information file (boa.env) (*1)
  - HTTP-IIOP gateway environment definition file (gwconfig) (*2)
  - Implementation Repository file (impl.db)(*1)
  - Initial Service file (init_svc/initial_services) (*1)
  - Queue control information file (queue_policy) (*1)
  - CORBA Service environment setup information file (odenvfile) (*1)
- Naming Service
  - Naming Service registration information file (file under the CosNaming directory) (*1)
  - Naming Service environment definition information file (nsconfig) (*1)
- Load balance function (Provided only by the Enterprise Edition)
  - Load balance function registration information file (file under the LBO directory) (*1)
  - Load balance environment definition file (nslbo.conf) (*1)
- Interface Repository
  - Interface Repository environment information file (irconfig, irpth) (*1)
  - Interface Repository data file (irobf.qfl, irobf.qfp, irobftran) (*1)

*1 For the locations where the files and directories are stored, refer to "Backing Up and Restoring Resources" - "Outline and Applicable Resources" in Maintenance (Resource Backup) in the Operator's Guide.

*2 For the locations where the files are stored, refer to 'gwconfig' in "CORBA Service Environment Definition" in the Tuning Guide.

These files may be exposed to the threat of unauthorized access from an ill-intentioned person.

To protect these files from this threat, make these files inaccessible by end users. For this purpose, it is recommended to allow access only by users having administrator authorization (superuser for a Solaris OE/Linux system, and Administrator for Windows(R) system).

Notes on Communication Data

There is a possible threat that an ill-intentioned person furtively reads communication data between the server and a user who has proper access permission. Another threat is that the data is altered and transmitted as the right data.

It is recommended to use SSL encryption to encrypt data for retaining security.

For information about SSL encryption, refer to Chapter 10, How to Use SSL with the CORBA Service.

Notes on the Port Number used by CORBA Service

CORBA Service uses port number 8002.

When this product is used in a DMZ, suppress requests from outside the 8002 port should use a security measure such as a firewall.

Notes on Creation and Operation of Java Applets

Be careful about the following points when creating and operating a Java applet that uses CORBA.

About Authorization Settings

If Java applets in operation are given more authorization than necessary, some malicious applets (including Javascripts) may use it to cause some problems on client machines, such as damaged files, leakage of data in files, leakage of individual user's information, and so on.

When you use Java applets, set only the minimum authorization that is required. Do not set authorizations other than those described in the following manuals:

- The Distributed Application Development Guide (CORBA Service Edition) (provided by Enterprise Edition and Standard Edition)
  - "Java Programming Guide" - "Execution of CORBA Applications" - "Client Setup (Pre-installed Java Clients)" - "Setting Permission for Java Libraries"
  - "Java Programming Guide" - "Execution of CORBA Applications" - "Client Setup (Portable-ORB)" - "Setting Permission for Java Libraries"
  - "Java Programming Guide" - "Digital Signatures in Applets" - "Digital Signature Procedures" - "policytool Command Setting (Supplements)"

About Errors and Exceptions

If information about an exception (stack trace) that occurs during operation of a Java applet is displayed on the screen (in a text field of the applet, on the Java console, etc.), internal information (internal structure) is leaked, which may be used by some malicious applets (including Javascripts).

It is recommended not to display exception information (stack trace).

Security Measures for Portable-ORB

Portable-ORB can be used with the following products:

- Interstage Application Server Enterprise Edition
- Interstage Application Server Standard Edition
- Interstage Application Server Plus

Unauthorized Access to Resource Files

Portable-ORB service has environment definition files as listed below:

- Portable-ORB environment definition file (config) (*1)
- Host information file (initial_hosts) (*1)
- Initial service file (initial_services) (*1)

*1 For the locations where the files are stored, refer to "Backing Up and Restoring Resources" - "Outline and Applicable Resources" in Maintenance (Resource Backup) in the Operator's Guide.

These files may be exposed to the threat of unauthorized access from an ill-intentioned person.

To protect these files from this threat, make these files inaccessible by end users. For this purpose, it is recommended to allow access only by users having administrator authorization (superuser for a Solaris OE/Linux system, and Administrator for Windows(R) system).

Notes on Communication Data

There is a possible threat that an ill-intentioned person furtively reads communication data between the server and a user who has proper access permission. Another threat is that the data is altered and transmitted as the right data.

It is recommended to use SSL encryption to encrypt data for retaining security.

Notes on Creation and Operation of Java Applet

Be careful about the following points when creating and operating a Java applet that uses Portable-ORB.

About Authorization Settings

If Java applets in operation are given more authorization than necessary, some malicious applets (including Javascript) may use it to cause problems on client machines, such as damaged files, leakage of data in files, leakage of individual user information, and so on.

When you use Java applets, set only the minimum authorization that is required. Do not set authorizations other than described in the following manuals:

- Distributed Application Development Guide (CORBA Service Edition) (provided by Enterprise Edition and Standard Edition)
  - "Java Programming Guide" - "Execution of CORBA Applications" - "Client Setup (Pre-installed Java Clients)" - "Setting Permission for Java Libraries"
  - "Java Programming Guide" - "Execution of CORBA Applications" - "Client Setup (Portable-ORB)" - "Setting Permission for Java Libraries"
  - "Java Programming Guide" - "Digital Signatures in Applets" - "Digital Signature Procedures" - "policytool Command Setting (Supplements)"

About Errors and Exceptions

If information about an exception (stack trace) that occurred during operation of a Java applet is displayed on the screen (in a text field of the applet, on the Java console, etc.), internal information (internal structure) is leaked, which may be used by some malicious applets (including Javascript).

It is recommended not to display exception information (stack trace).

Security Measures for Event Service

Event service can be used with the following products:

- Interstage Application Server Enterprise Edition
- Interstage Application Server Standard Edition
- Interstage Application Server Plus

Unauthorized Access to Resource Files

Event service has environment definition files as listed below:

- Event Service configuration information (essystem.cfg) (*1)
- Event channel operating environment (esgrpX.grp) (*1)
- Event channel group management information (esmnggrp.db) (*1)
- Unit definition file (file with the def extension) (*1)
- Log file (ESLOG.log) (*2)

  *1 For the locations where the files are stored, refer to "Backing Up and Restoring Resources" - "Outline and Applicable Resources" in Maintenance (Resource Backup) in the Operator's Guide.

  *2 For the locations where the files are stored, refer to "Messages Output by the Event Service" in Messages Output to a Log File in the Messages.

If a unit definition file is set during unit creation (when the 'esmkunit' command is executed) for nonvolatile operation, the following directories are maintained:

- Directory to store a transaction file specified by "trandir."
- Directory to store a system file specified by "sysdir."
- Directory to sore an event data file specified by "usedir."

These files and directories may be exposed to the threat of unauthorized access from an ill-intentioned person.

To protect these files and directories from this threat, make these files and directories inaccessible by end users. For this purpose, it is recommended to allow access only to users having administrator authorization (superuser for a Solaris OE/Linux system, and Administrator for Windows(R) system).

Security Measures for IJServer Operation

IJServer is an operating environment for JEEE applications.

Unauthorized Access to Resource Files

When IJServer is operated, the resource files for IJServer are stored in the ijserver directory under the J2EE common directory. These files may be subjected to unauthorized access by malicious persons or machines.

To protect these files from such threats, access to the files from general users can be inhibited. It is recommended to permit access only by users with administrator authority (administrator of the Windows(R) system).

Notes on IJServer Execution

IJServer is an operating environment for J2EE applications and IJServer itself is executed as a process. Only users with administrator authority (administrators of the Windows(R) system) can execute IJServer. It is recommended to carefully select the users to whom administrator authority is assigned and periodically review this to improve safety.

Security Measures Concerning Operation of Smart Repository

Smart Repository can be used with the following products:

- Interstage Application Server Enterprise Edition
- Interstage Application Server Standard Edition
- Interstage Application Server Plus

About Operation

Take the following measures to prevent incorrect use during operation.

- Users who are well informed about the overall information system including Smart Repository must perform Smart Repository operation and they must receive appropriate training.
- Smart Repository must always be correctly controlled and operated according to the rules established in the manuals.

Blocking External Access

Set up a firewall and routers appropriately, prevent the intrusion of unauthorized external packets and inhibit access to ports other than those specified.

Restriction of Services

By restricting remotely accessible services (such as telnet and ftp) on nodes where Interstage is operating, you can prevent unauthorized accesses. This measure is effective against unauthorized accesses made through networks.

For details of how to restrict such remotely accessible services, refer to the manual for each platform.

Notes on Accessing the Smart Repository Server

When an LDAP client accesses the Smart Repository server, there is a risk that an ill-intentioned person on the network may access the Smart Repository server by impersonating a user having proper access permissions. SSL encryption using SSL version 3 (client authentication) is recommended.

For SSL communication details, refer to "Method for Using SSL in Smart Repository."

Security Measures for Fujitsu Enabler

This section explains how to configure the security settings for the Fujitsu Enabler account.

Account Used for Fujitsu Enabler

Windows

Fujitsu Enabler is used for Smart Repository.

For Fujitsu Enabler, an account "oms" is used for a special purpose. To prevent malicious access using the oms account, inhibit the local login with the oms account.

Implement a similar setting for the account that has been specified for the installation of the Exchange service. If the computer is a domain controller, set up the domain policy instead of the local policy.

1. Open the [Control Panel] window from the [Start] menu, and select [Local Security Policy] from [Administration Tool].
2. Open [Local Policy], and click on [Allocate User Permission].
3. In the frame on the right, double-click [Reject Local Logon].
4. In the [Local Security Policy Setting] dialog box, click the [Add] button.
5. In the [User or Group Selection] dialog box, select the oms account, click the [Add] button and then click the [OK] button.
6. Confirm that a check mark is placed in the [Set the local policy] check box for the oms account.
7. Click the [OK] button.

If the local logon cannot be rejected because the oms account has been used for another purpose, use the password protection for the screen saver to reduce security problems.

1. Log on with the oms account.
2. Right-click on the desktop to display the [Display Property] dialog box.
3. Select the [Screen Saver] tab.
4. From the drop down menu, select any screen saver and place a check in the [Use password protection] check box.
5. Click the [OK] button.

Measures for Multi server Management

This section explains the use of "roles" in Multi server Management.

Security Role Settings

When using Multi Server Management, it is important that the authority set for a user to log in to the Interstage Management Console is appropriate. This user authority is called a “role”.

The executable operations vary according to the role authority. For further information and details about the role, refer to “Login User Authentication” in the Operator’s Guide.

Measures for Configuration Manager

This section explains the security measures for the Configuration Manager.

Illegal Access to Resource Files

The Configuration Manager uses the following files.

- Business configuration management function repository (Note 1)

(Note 1) This is the folder that is set in “Repository Environment Settings” of the Interstage Management Console [Configuration Management] tab.

It is possible that these files may be exposed to the threat of illegal access by a person (or machine) with malicious intent.

One measure to reduce this threat is to prevent access by general users. To achieve this, it is recommended that settings are made for these files so that only a user with administrator authority can access them (in Solaris OE/Linux systems, this is a super user, and in Windows(R) systems it is the Administrator).

Part II

Authentication and Access Control

Chapter 3

Authentication and Access Control for the Interstage HTTP Server

This chapter describes the authentication and access control that Interstage HTTP Server provides.

Types of Authentication

There are three types of authentication, as shown below.

- User authentication (Basic Authentication)
- IP access control

Note

User authentication and IP access control can be used independently or together.

By using the online check function, it is possible to store the user name, password, and group information used for user authentication on the directory server and check the user name/password online.

User Authentication (Basic Authentication)

User authentication controls user names and passwords. User authentication has the function to limit access to the resource on the Web server for each user by controlling user names and passwords.

Web server determines whether or not to permit access to the resource from user names and passwords entered in the web browser.

Figure 3-1 shows the function of user authentication.

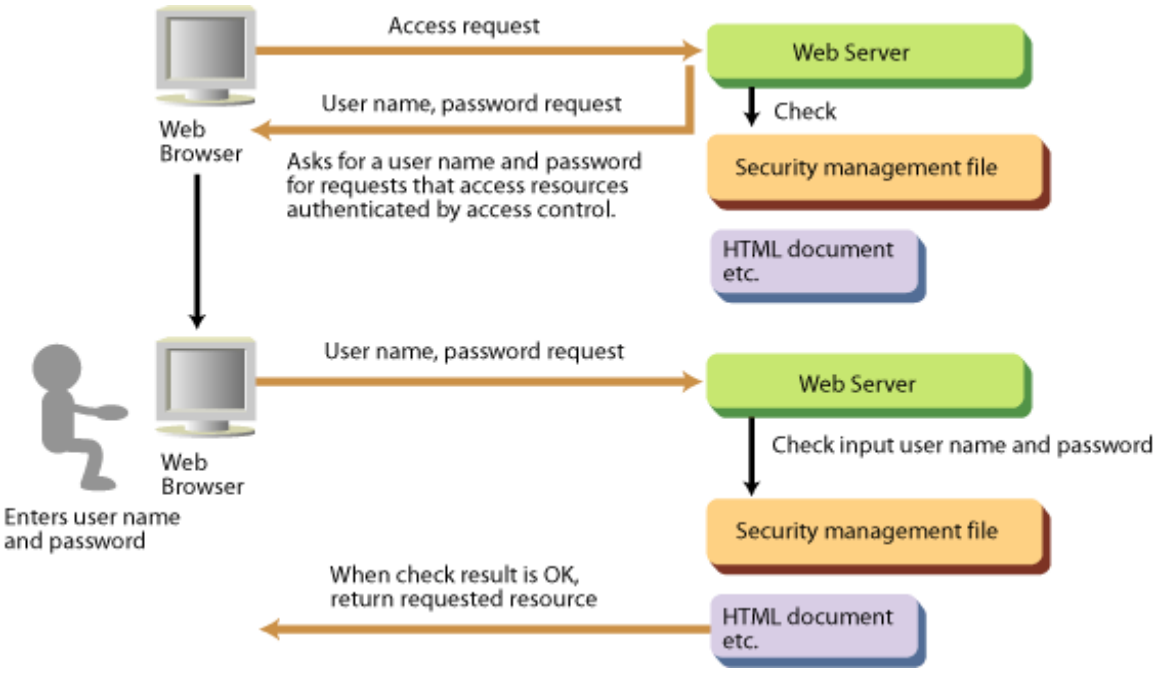

Access request
Web Server
User name, password request
Check
Web Browser
Asks for a user name and password for requests that access resources authenticated by access control.
Security management file
HTML document etc.
User name, password request
Web Server
Check input user name and password
Web Browser
Enters user name and password
Security management file
When check result is OK, return requested resource
HTML document etc.

Figure 3-1 User Authentication

Remarks

When SSL is used between the client and the server for user authentication, the user name and the password are encrypted, which makes them almost impossible to intercept or steal.

IP Access Control

IP access control limits accessing the resource on the Web server for each IP address of the equipment in the access source.

Web server determines whether or not to permit access to the resource from the IP address of the machine that is attempting access.

Figure 3-2 shows the function of IP access control.

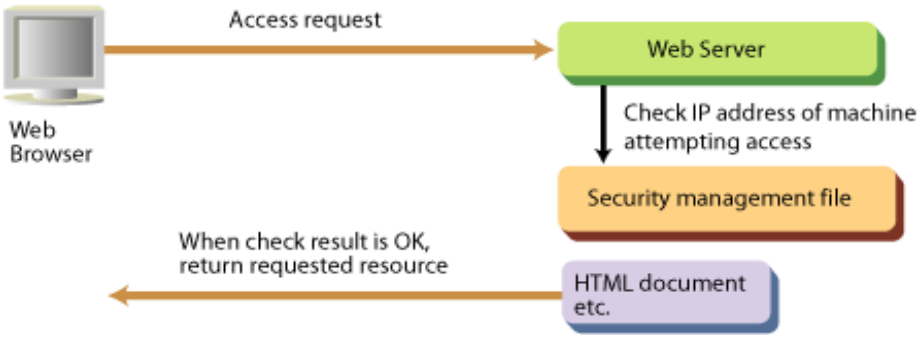


Figure 3-2 IP Access Control

Online Collation

This function is used to control and store the user names, passwords and group information used in user authentication in the directory server.

Web server uses the LDAP (Lightweight Directory Access Protocol) V3 to communicate with the directory server and collates the user name/password in the online operation.

When this function is used, the directory server centrally controls the user names/passwords so that the user names/passwords that are common with Web server can be used.

Figure 3-3 shows online collation.

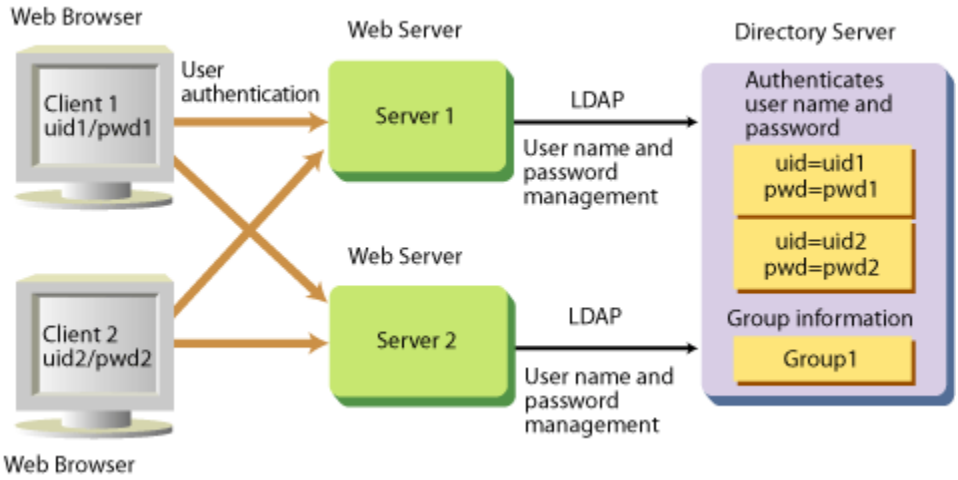


Figure 3-3 Online Collation

Note

When the online check function is used, the authentication function of Web server cannot be used.

Setting the User Authentication

User authentication is set according to the following procedures.

1. Registering a user password
2. Editing the environment definition file

Note

When the online collation function is in use user authentication cannot be used.

Registering a User Password

Register a password for users to whom access permission is to be provided in the password file, by executing the htpasswd command after the command prompt.

Example Windows

To create a new password file 'C:\Interstage\F3FMihs\conf\password.txt' and register a password for user 'user1':

```
C:\Interstage\bin\htpasswd -c C:\Interstage\F3FMihs\conf\password.txt user1
```

Example Solaris OE Linux

To create a new password file '/opt/FJSVihs/conf/password.txt' and register a password for user 'user1':

```
/opt/FJSVihs/bin/htpasswd -c /opt/FJSVihs/conf/password.txt user1
```

Notes

- To register the second and later users or to change a user password already registered, the -c option does not need to be specified for the htpasswd command.
- To delete a user password, edit the password file using a text editor. In the password file, the user names and passwords are coded in the format shown below on a text editor. To delete the user password of 'user2,' delete the entire line of 'user2'.

```
user1:$apr1$SR3.....$4aQAE2EU9NZTtbkxMEOa4/
user2:$apr1$DS3.....$tEb4EYLhraAc1p2wIygTV/
```

Editing the Environment Definition File

To allow the users whose password has been registered in the password file to access directories under a specified directory, use the following directives in the environment definition file (httpd.conf) of Interstage HTTP Server. By doing this, names and passwords of users who make access requests from Web browsers are checked and any access attempts from users whose names and passwords have not been registered in the password file are rejected.

Example Windows

To allow the users whose names and passwords have been registered in the password file "C:\Interstage\F3FMihs\conf\password.txt" to access directories under a specified directory "C:\Interstage\F3FMihs\htdocs\users\name":

```
# Directory
<Directory "C:/Interstage/F3FMihs/htdocs/users/name">
  # Password file
  AuthUserFile "C:/Interstage/F3FMihs/conf/password.txt"
  # Title displayed on the authentication window
  AuthName "Secret directory"
  # Type of authentication
  AuthType Basic
  # Rule to be applied for user authentication
  Require valid-user
</Directory>
```

Example Solaris OE Linux

To allow the users whose names and passwords have been registered in the password file "/opt/FJSVihs/conf/password.txt" to access directories under a specified directory "/opt/FJSVihs/htdocs/users/name":

```
# Directory
<Directory "/opt/FJSVihs/htdocs/users/name">
  # Password file
  AuthUserFile "/opt/FJSVihs/conf/password.txt"
  # Title displayed on the authentication window
  AuthName "Secret directory"
  # Type of authentication
  AuthType Basic
  # Rule to be applied for user authentication
  Require valid-user
</Directory>
```

Note

When user authentication is set for the Servlet service application URL, the <Directory> section of the above example cannot be used. Use the <Location> section.

Relating Directives

- AuthName
- AuthType
- AuthUserFile
- <Directory>
- Require

Relating Directives

When user authentication is used, the following directives are related to settings of the environment definition file.

The description includes the following:

Name

Directive name

Synopsis

Directive format

Description

Functional overview of the directive

Context

Directive-set location indicated with one of the following keywords:

Global context

Setting used for action of the entire Web server

Virtual host

Setting which is available in the <VirtualHost> section and used for action of the virtual host

Directory

Setting which is available in the <Directory>, <Location>, and <Files> sections and used for action in response to a request for a specified directory, URL, or file

Default Value

Value assumed when the directive is omitted. If a directive indicated with 'None' is omitted, the directive function is disabled.

Note

Notes on the use of the directive

Examples

Directive example (included only for a directive which requires complicated setting).

AuthName

Name

AuthName

Synopsis

AuthName 'title'

Description

Specifies the title displayed on the authentication screen in the ASCII alphanumeric characters (1 byte characters).

Context

Directory

Default Value

None

AuthType

Name

AuthType

Synopsis

AuthType Basic

Description

Specifies the type of authentication 'Basic'.

Basic

Sets basic authentication (the passwords are plain text).

Context

Directory

Default Value

None

Note

Digest authentication cannot be used in Interstage HTTP Server.

AuthUserFile

Name

AuthUserFile

Synopsis

AuthUserFile file-name

Description

Specifies the name of the password file used for user authentication (the name of the text file that includes the list of users and their passwords).

Context

Directory

Default Value

None

Module

mod_auth

<Directory>

Name

<Directory>

Synopsis

<Directory directory-path> ... </Directory>

Description

Specifies the directory section only when a directive is used within the specific directory and sub-directories of that directory.

The directory name can be specified using a relative path, wild card (? indicates a specific character, * indicates a character string), and regular expressions.

Within the specified directory, all the directives allowed in the directory context can be used.

Context

Global context, Virtual host

Default Value

None

Require

Name

Require

Synopsis

Require valid-user|user user-name|group group-name

Description

Specifies the rule to be applied for user authentication.

valid-user

Authenticates all valid users.

When the online collation function is used, users registered with the directory server are allowed.

user user-name

Authenticates users specified by *user-name*.

When the online collation function is used, the uid attribute of the user is specified as the *user-name*.

Use a space as a delimiter between user and *user-name*.

group group-name

Authenticates groups specified by *group-name*.

When the online collation function is used, the cn attribute of the group is specified as the *group-name*.

Use a space as a delimiter between group and *group-name*.

Context

Directory

Default Value

None

Examples

To authenticate a user 'taro':

```
Require user taro
```

To allow authentication of a user belonging to the group entry with the cn attribute 'ihsgroup' when the online collation function is used:

```
Require group cn=ihsgroup,ou=User,ou=interstage,o=fujitsu,dc=com
```

Setting the IP Access Control

For IP access control, you can allow only specified hosts to make access to directories under a specified directory using the following directives in the environment definition file (httpd.conf) of Interstage HTTP Server. By doing this, any access from Web browsers that are on unspecified hosts are rejected.

Example Windows

To allow a specified host '192.168.1.1' to access directories under a specified directory "C:\Interstage\F3FMihs\htdocs\secret":

```
# Directory
<Directory "C:/Interstage/F3FMihs/htdocs/secret">
    # Specify the order in which the directives are applied
    Order deny,allow
    # Specify the access which is prohibited
    Deny from all
    # Specify the access which is allowed
    Allow from 192.168.1.1
</Directory>
```

Example Solaris OE Linux

To allow a specified host '192.168.1.1' to access directories under a specified directory "/opt/FJSVihs/htdocs/secret":

```
# Directory
<Directory "/opt/FJSVihs/htdocs/secret">
    # Specify the order in which the directives are applied
    Order deny,allow
    # Specify the access which is prohibited
    Deny from all
    # Specify the access which is allowed
    Allow from 192.168.1.1
</Directory>
```

Note

When IP access control is set for the Servlet service application URL, the <Directory> section of the above example cannot be used. Use the <Location> section.

Relating Directives

- Allow
- Deny
- <Directory>
- Order

Relating Directives

When IP access control is used, the following directives are related to settings of the environment definition file.

The description includes the following:

Name

Directive name

Synopsis

Directive format

Description

Functional overview of the directive

Context

Directive-set location indicated with one of the following keywords:

Global context

Setting used for action of the entire Web server

Virtual host

Setting which is available in the <VirtualHost> section and used for action of the virtual host

Directory

Setting which is available in the <Directory>, <Location>, and <Files> sections and used for action in response to a request for a specified directory, URL, or file

Default Value

Value assumed when the directive is omitted. If a directive indicated with 'None' is omitted, the directive function is disabled.

Module

Name of the module that implements the directive function. A directive with no module name indication is included in the basic module.

Allow

Name

Allow

Synopsis

Allow from host|network[/mask] [host|network[/mask]] ...

Description

Specifies a host or network that is granted access to the directories.

Specifying 'all' for the host entry allows all hosts to access the directories. Specifying the IP address of a host allows only that host to access the directories.

Context

Directory

Default Value

None

Module

mod_access

Deny

Name

Deny

Synopsis

Deny from host|network[/mask] [host|network[/mask]] ...

Description

Specifies a host or network that is denied access to the directories.

Specifying 'all' to the host entry, denies access to all hosts. Specifying the IP address of a host denies access to the directories for that host only.

Context

Directory

Default Value

None

Module

mod_access

<Directory>

Name

<Directory>

Synopsis

<Directory directory-path> ... </Directory>

Description

Specifies the directory section only when a directive is used within the specific directory and sub-directories of that directory.

The directory name can be specified using a relative path, wild card (? indicates a specific character, * indicates a character string), and regular expressions.

Within the specified directory, all the directives allowed in the directory context can be used.

Context

Global context, Virtual host

Default Value

None

Order

Name

Order

Synopsis

Order order

Description

Specifies the order in which the Allow directive and the Deny directive are applied. Specifying 'deny,allow' applies the Deny directive first, then the Allow directive.

Context

Directory

Default Value

deny,allow

Module

mod_access

Setting the Online Collation Function

Set the operation of the online collation function according to the following procedure.

The directory servers used when the online access management function is operated, are intended for the following services:

- Smart Repository

With the online collation function, operation with SSL enabled or disabled can be set between Interstage HTTP Server and the directory server. For operation with SSL enabled, the procedure for configuring environments varies depending on whether Interstage HTTP Server and the directory server are on the same system. Configure environments using the table below as a reference.

Table 3-1 Environment Configuration

	SSL environment configuration for Smart Repository	SSL environment configuration for Interstage HTTP Server (See Note 1)	Refer to Procedure
Operation Without Using SSL	Configuration is not required.	Configuration is not required.	Configuration procedure 1
Operation Using the SSL (when the servers are on the same system)	Configuration is required.	Configuration is not required. However, there is no problem if configuration is performed.	Configuration procedure 2
Operation Using the SSL (when the servers are on different systems)	Configuration is required.	Configuration is required.	Configuration procedure 3

Note 1

A certificate/key management environment configured with the SMEE commands can be used as an SSL environment for Interstage HTTP Server. Set either of the SSL environments according to the operation type.

Notes

- To use the online collation function, the client API library (InfoDirectory SDK) is required in the same machine as the Interstage HTTP Server.
- When the online collation function is used, user authentication cannot be used.

The following procedure is for operating the online collation function.

Operation without Using SSL

Configuration Procedure 1

This section explains the procedure for operating the online collation function without using the SSL between the Interstage HTTP Server and directory server.

1. Set up the environment of the directory server

   Set the directory server environment. Refer to Setting the Directory Server Environment for details of how to set the directory server environment.

2. Set the Interstage HTTP Server environment definition file (httpd.conf).

   Refer to Set the Interstage HTTP Server Environment Definition File for details of how to set the Interstage HTTP Server environment definition file (httpd.conf).

Operation Using SSL

Configuration Procedure 2 (when Interstage HTTP Server and Smart Repository are on the same system)

The following procedure is for operating the online collation function, with SSL enabled between Interstage HTTP Server and the directory server; when Interstage HTTP Server and Smart Repository are on the same system.

1. Set up the environment of the directory server

   Set the directory server environment. An SSL environment must also be set on the directory server. (Note 1) Refer to Setting the Directory Server Environment for details of how to set the directory server environment.

2. Set the Interstage HTTP Server environment definition file (httpd.conf).

   Refer to Set the Interstage HTTP Server Environment Definition File for details of how to set the Interstage HTTP Server environment definition file (httpd.conf).

Note 1

To register users in the Interstage certificate environment, set users who have been specified in the User directive of the Interstage HTTP Server environment definition file (httpd.conf), if you set the SSL environment of Smart Repository on Solaris OE/Linux.

Configuration Procedure 3 (when Interstage HTTP Server and Smart Repository are on different systems)

The following procedure is for operating the online collation function with SSL enabled between Interstage HTTP Server and the directory server; when Interstage HTTP Server and Smart Repository are on different systems.

1. Set up the environment of the directory server

   Set the directory server environment. An SSL environment must also be set on the directory server. Refer to Setting the Directory Server Environment for details of how to set the directory server environment.

2. Set up the SSL environment of Interstage HTTP Server

   When you use SSL of the Interstage certificate environment

   The Interstage certificate environment is configured. For details, refer to Configuring Environments in Chapter 7 of the Security Manual. Create an Interstage certificate environment owner group for Solaris OE/Linux.

   When you use SSL of the certificate/key management environment configured with the SMEE command (Refer to Note 2)

   1) Create a certificate/key management environment.

      For details, refer to Creating a Certificate/Key Management Environment in Chapter 8, Setting and Use of the Certificate/Key Management Environment Using the SMEE Command.

   2) Create a secret key and acquire a certificate.

      For details, refer to Creating a Secret Key and Acquiring a Certificate in Environment in Chapter 8, Setting and Use of the Certificate/Key Management Environment Using the SMEE Command.

   3) Register the certificate and CRL.

      For details, refer to Registering the Certificate and CRL Environment in Chapter 8, Setting and Use of the Certificate/Key Management Environment Using the SMEE Command.

3. Set the Interstage HTTP Server environment definition file (httpd.conf).

   Set the Interstage HTTP Server environment definition file (httpd.conf). Refer to Set the Interstage HTTP Server Environment Definition File for details on the setting process.

Note 2

In the Solaris OE/Linux system, a user other than the super-user authority needs to execute. (The user other than the super-user authority set the process of Web Server for consideration on security.)

Specify the user and group in the User and Group directives in the environment definition file (httpd.conf) in step 3.

Setting the Directory Server Environment

To use the online collation function, the environment of the directory server need be set up.

The following operations are required for setting up the directory server environment:

1. Preparing the Directory Server
2. Creating Entries

Preparing the Directory Server

Prepare the directory server and generate a repository (DSA). To perform secure communication between Interstage HTTP Server and directory server using the SSL, also the SSL environment needs to be set up in the directory server.

For the directory server, refer to the Smart Repository Operator's Guide.

Creating Entries

On the directory server, create entries for user and group information with an entry management tool.

If user authentication is set with a group name, the Interstage HTTP Server environment definition file need not be edited even when user information is changed.

Creating User Entry

Create the user entry with the following inetOrgPerson object class.

For the user entry, the following items must be set.

Table 3-2 User Entry Settings

Item	Description
cn attribute	Sets a user entry name.
uid attribute	Sets a user name for online collation.
userPassword attribute	Sets a password associated with the user name.

Example of User Entry Configuration

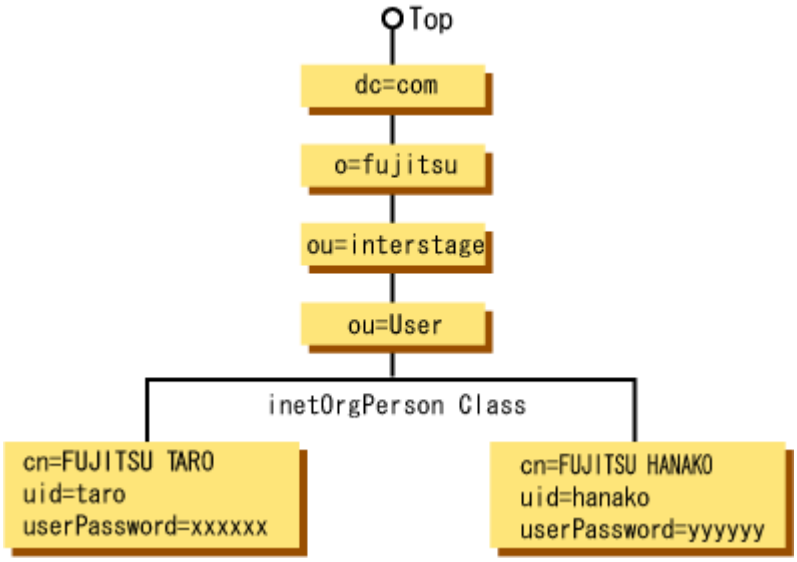


Figure 3-4 Creating User Entry

Creating Group Entry

Create the group entry with the following groupOfNames object class.

For the group entry, the following items must be set.

Table 3-3 Group Entry Settings

Item	Description
cn attribute	Sets a name of a group to which the user performing online collation belongs.
member attribute	Sets a DN name of the user belonging to a group.

Example of Group Entry

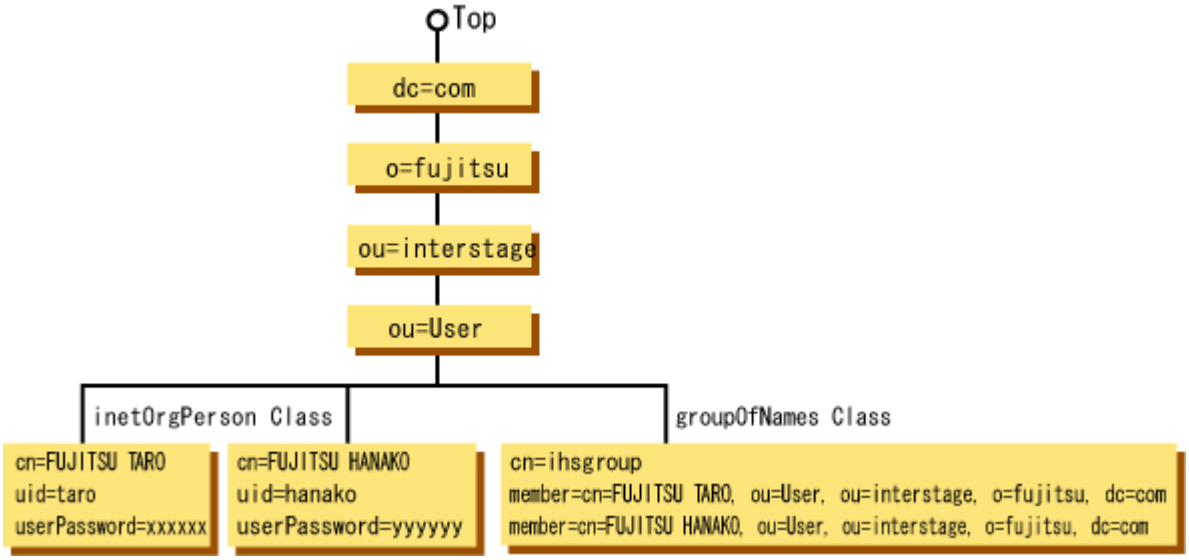


Figure 3-5 Group Entry Configuration

Set the Interstage HTTP Server Environment Definition File

Define the online collation function according to the mode of operation in the environment definition file (httpd.conf) for the Interstage HTTP Server.

The method for setting the environment definition file for Interstage HTTP Server (httpd.conf) varies depending on whether operation is set with SSL disabled or enabled between Interstage HTTP Server and the directory server. Set the file using the table below as a reference.

Table 3-4 Environment Definition File Settings

SSL environment used by Interstage HTTP Server	Refer to Setting Example
Operation without using SSL	Setting 1
Operation using the SSL of the Interstage certificate environment (Note)	Setting 2
Operation using the SSL of the certificate/key management environment configured with the SMEE command	Setting 3

Note

This also applies when SSL environments configured on Smart Repository are used.

An example of definition in the Interstage HTTP Server environment definition file (httpd.conf) for each case is shown below.

Setting 1: Operation without Using SSL

Example Windows

Running the online collation function without using SSL, under the following settings:

Directory server 'hostname'

Port number '389'

BindDN name used to access the directory server 'cn=manager,ou=interstage,o=fujitsu,dc=com'

Name of the tree containing user information on the directory server 'ou=User,ou=interstage,o=fujitsu,dc=com'

```
# Add the module (Delete the comment)
LoadModule mod_ldap_module modules/mod_ldap.dll
AddModule mod_ldap.c

# Directory
<Directory "C:/Interstage/F3FMihs/htdocs/securityzone">

  # BindDN name used to access the directory server
  AuthLDAPBindDN         cn=manager,ou=interstage,o=fujitsu,dc=com
  # Password for the BindDN-name
  AuthLDAPBindPassword   password
  # Specify whether to enable LDAP authentication (on: enable, off: disable).
  AuthLDAPEnabled        on

  # Title displayed on the authentication window
  AuthName               "title"
  # Basic authentication
  AuthType               Basic
  # Name of the host running the directory server
  AuthLDAPHost           hostname
  # Port number of the directory server
  # (389:optional value for not using SSL, 636:optional value for using SSL)
  AuthLDAPPort           389
  # Name of the tree containing user information on the directory server
  AuthLDAPbasedn         ou=User,ou=interstage,o=fujitsu,dc=com

  # Rule to be applied for LDAP authentication
  Require                valid-user

  # Specify whether to enable SSL (off: disable, on: enable).
  AuthLDAPSecure         off

</Directory>
```

Example Solaris OE Linux

Running the online collation function without using SSL, under the following settings:

Directory server 'hostname'

Port number '389'

BindDN name used to access the directory server 'cn=manager,ou=interstage,o=fujitsu,dc=com'

Name of the tree containing user information on the directory server 'ou=User,ou=interstage,o=fujitsu,dc=com'

```
# Add the module (Delete the comment)
LoadModule mod_ldap_module libexec/mod_ldap.so
AddModule mod_ldap.c

# Directory
<Directory "/opt/FJSVihs/htdocs/securityzone">

  # BindDN name used to access the directory server
  AuthLDAPBindDN         cn=manager,ou=interstage,o=fujitsu,dc=com
  # Password for the BindDN-name
  AuthLDAPBindPassword   password
  # Specify whether to enable LDAP authentication (on: enable, off: disable).
  AuthLDAPEnabled        on

  # Title displayed on the authentication window
  AuthName               "title"
  # Basic authentication
  AuthType               Basic
  # Name of the host running the directory server
  AuthLDAPHost           hostname
  # Port number of the directory server
  # (389:optional value for not using SSL, 636:optional value for using SSL)
  AuthLDAPPort           389
  # Name of the tree containing user information on the directory server
  AuthLDAPbasedn         ou=User,ou=interstage,o=fujitsu,dc=com

  # Rule to be applied for LDAP authentication
  Require                valid-user

  # Specify whether to enable SSL (off: disable, on: enable).
  AuthLDAPSecure         off

</Directory>
```

Setting 2: Operation Using the SSL (setting for using an Interstage certificate environment or for using SSL configured on Smart Repository)

Example Windows

Running the online collation function without using SSL, under the following settings:

Directory server 'hostname'

Port number '636'

BindDN name used to access the directory server 'cn=manager,ou=interstage,o=fujitsu,dc=com'

Name of the tree containing user information on the directory server 'ou=User,ou=interstage,o=fujitsu,dc=com'

```
# Add the module (Delete the comment)
LoadModule mod_ldap_module modules/mod_ldap.dll
AddModule mod_ldap.c

# Directory
<Directory "C:/Interstage/F3FMihs/htdocs/securityzone">

  # BindDN name used to access the directory server
  AuthLDAPBindDN         cn=manager,ou=interstage,o=fujitsu,dc=com
  # Password for the BindDN-name
  AuthLDAPBindPassword   password
  # Specify whether to enable LDAP authentication (on: enable, off: disable).
  AuthLDAPEnabled        on

  # Title displayed on the authentication window
  AuthName               "title"
  # Basic authentication
  AuthType               Basic
  # Name of the host running the directory server
  AuthLDAPHost           hostname
  # Port number of the directory server
  # (389:optional value for not using SSL, 636:optional value for using SSL)
  AuthLDAPPort           636
  # Name of the tree containing user information on the directory server
  AuthLDAPbasedn         ou=User,ou=interstage,o=fujitsu,dc=com

  # Rule to be applied for LDAP authentication
  Require                valid-user

  # Specify whether to enable SSL (off: disable, on: enable).
  AuthLDAPSecure         on

</Directory>
```

Example Solaris OE Linux

Running the online collation function without using SSL, under the following settings:

Directory server 'hostname'

Port number '636'

BindDN name used to access the directory server 'cn=manager,ou=interstage,o=fujitsu,dc=com'

Name of the tree containing user information on the directory server 'ou=User,ou=interstage,o=fujitsu,dc=com'

User who registered to owner groups of Interstage certificate environment 'nobody'

Group to which the above user belongs: 'nobody'

```
# Add the module (Delete the comment)
LoadModule mod_ldap_module libexec/mod_ldap.so
AddModule mod_ldap.c

# Set the user who registered to owner groups of Interstage certificate
environment
User nobody
# Set the group to which the above user belongs
Group nobody

# Directory
<Directory "/opt/FJSVihs/htdocs/securityzone">

  # BindDN name used to access the directory server
  AuthLDAPBindDN         cn=manager,ou=interstage,o=fujitsu,dc=com
  # Password for the BindDN-name
  AuthLDAPBindPassword   password
  # Specify whether to enable LDAP authentication (on: enable, off: disable).
  AuthLDAPEnabled        on

  # Title displayed on the authentication window
  AuthName               "title"
  # Basic authentication
  AuthType               Basic
  # Name of the host running the directory server
  AuthLDAPHost           hostname
  # Port number of the directory server
  # (389:optional value for not using SSL, 636:optional value for using SSL)
  AuthLDAPPort           636
  # Name of the tree containing user information on the directory server
  AuthLDAPbasedn         ou=User,ou=interstage,o=fujitsu,dc=com

  # Rule to be applied for LDAP authentication
  Require                valid-user

  # Specify whether to enable SSL (off: disable, on: enable).
  AuthLDAPSecure         on

</Directory>
```

Setting 3: Operation Using the SSL (setting for using a certificate/key management environment configured with the SMEE commands)

Example Windows

Running the online collation function using the SSL, under the following settings:

Directory server 'hostname'

Port number '636'

BindDN name used to access the directory server 'cn=manager,ou=interstage,o=fujitsu,dc=com'

Name of the tree containing user information on the directory server 'ou=User,ou=interstage,o=fujitsu,dc=com'

Slot information directory 'D:\sslenv\slot'

Operation control directory 'D:\sslenv\sslcert'

Token label 'token01'

User PIN 'userpin'

```
# Add the module (Delete the comment)
LoadModule mod_ldap_module modules/mod_ldap.dll
AddModule mod_ldap.c

# Directory
<Directory "C:/Interstage/F3FMihs/htdocs/securityzone">

  # BindDN name used to access the directory server
  AuthLDAPBindDN         cn=manager,ou=interstage,o=fujitsu,dc=com
  # Password for the BindDN-name
  AuthLDAPBindPassword   password
  # Specify whether to enable LDAP authentication (on: enable, off: disable).
  AuthLDAPEnabled        on

  # Title displayed on the authentication window
  AuthName               "title"
  # Basic authentication
  AuthType               Basic
  # Name of the host running the directory server
  AuthLDAPHost           hostname
  # Port number of the directory server
  # (389:optional value for not using SSL, 636:optional value for using SSL)
  AuthLDAPPort           636
  # Name of the tree containing user information on the directory server
  AuthLDAPbasedn         ou=User,ou=interstage,o=fujitsu,dc=com

  # Rule to be applied for LDAP authentication
  Require                valid-user

  # Specify whether to enable SSL (off: disable, on: enable).
  AuthLDAPSecure         on
  # Slot information directory
  AuthLDAPSlotPath       "D:\sslenv\slot"
  # Operation control directory
  AuthLDAPCertPath       "D:\sslenv\sslcert"
```

```
  # Token label
  AuthLDAPTknLbl         token01
  # User PIN file
  AuthLDAPTknPwd         userpin

</Directory>
```

Example Solaris OE Linux

Running the online collation function using the SSL, under the following settings:

Directory server 'hostname'

Port number '636'

BindDN name used to access the directory server 'cn=manager,ou=interstage,o=fujitsu,dc=com'

Name of the tree containing user information on the directory server
'ou=User,ou=interstage,o=fujitsu,dc=com'

Slot information directory '/home/slot'

Operation control directory '/home/sslcert'

Token label 'token01'

User PIN 'userpin'

User set up certificate and key management environment 'user1'

Group to which the above user belongs: 'group1'

```
# Add the module (Delete the comment)
LoadModule mod_ldap_module libexec/mod_ldap.so
AddModule mod_ldap.c

# Set the user set up certificate and key management environment
User user1
# Set the group to which the above user belongs
Group group1

# Directory
<Directory "/opt/FJSVihs/htdocs/securityzone">

  # BindDN name used to access the directory server
  AuthLDAPBindDN         cn=manager,ou=interstage,o=fujitsu,dc=com
  # Password for the BindDN-name
  AuthLDAPBindPassword   password
  # Specify whether to enable LDAP authentication (on: enable, off: disable).
  AuthLDAPEnabled        on

  # Title displayed on the authentication window
  AuthName               "title"
  # Basic authentication
  AuthType               Basic
  # Name of the host running the directory server
  AuthLDAPHost           hostname
  # Port number of the directory server
```

```
  # (389:optional value for not using SSL, 636:optional value for using SSL)
  AuthLDAPPort           636
  # Name of the tree containing user information on the directory server
  AuthLDAPbasedn         ou=User,ou=interstage,o=fujitsu,dc=com

  # Rule to be applied for LDAP authentication
  Require                valid-user

  # Specify whether to enable SSL (off: disable, on: enable).
  AuthLDAPSecure         on
  # Slot information directory
  AuthLDAPSlotPath       "/home/slot"
  # Operation control directory
  AuthLDAPCertPath       "/home/sslcert"
  # Token label
  AuthLDAPTknLbl         token01
  # User PIN file
  AuthLDAPTknPwd         userpin

</Directory>
```

Note

When the online collation function is set for the Servlet service application URL, the <Directory> section of the above example cannot be used. Use the <Location> section.

Relating Directives

- AddModule
- AuthLDAPbasedn
- AuthLDAPBindDN
- AuthLDAPBindPassword
- AuthLDAPCertPath
- AuthLDAPEnabled
- AuthLDAPHost
- AuthLDAPPort
- AuthLDAPSecure
- AuthLDAPSlotPath
- AuthLDAPTknLbl
- AuthLDAPTknPwd
- AuthName
- AuthType

- <Directory>
- Group
- LoadModule
- Require
- User

Relating Directives

The following directives are related to settings of the environment definition file to use the online collation function.

The description includes the following:

Name

Directive name

Synopsis

Directive format

Description

Functional overview of the directive

Context

Directive-set location indicated with one of the following keywords:

Global context

Setting used for action of the entire Web server

Virtual host

Setting which is available in the <VirtualHost> section and used for action of the virtual host

Directory

Setting which is available in the <Directory>, <Location>, and <Files> sections and used for action in response to a request for a specified directory, URL, or file

Default Value

Value assumed when the directive is omitted. If a directive indicated with 'None' is omitted, the directive function is disabled.

Initial value

Initial directive value

Module

Name of the module that implements the directive function. A directive with no module name indication is included in the basic module.

Note

Notes on the use of the directive

Examples

Directive example (included only for a directive which requires complicated setting).

AddModule

Name

AddModule

Synopsis

AddModule module [module] ...

Description

Enables read modules or embedded modules.

Context

Global context

Default Value

None

AuthLDAPbasedn

Name

AuthLDAPbasedn

Synopsis

AuthLDAPbasedn BaseDN-name

Description

Specifies the name of the tree that is storing information about users in the directory server using the DN name.

When information about the users is stored in multiple directories, specify the name of a high-order DN which is inclusive of all the user information storing directories. The directory specified in BaseDN is handled as the top directory from which a search is made for information about the users. The character string specified in BaseDN is transferred to the directory server as it is; specify it using the code used on the directory server.

BaseDN-name

Use ASCII characters (1-byte characters: 0 to 9, A to Z, and a to z). Up to 256 bytes are allowed.

Context

Directory

Default Value

None

Module

mod_ldap

AuthLDAPBindDN

Name

AuthLDAPBindDN

Synopsis

AuthLDAPBindDN BindDN-name

Description

Specifies the BindDN name used for access to the directory server. When making anonymous access, omit this directive.

BindDN-name

Use ASCII characters (1-byte characters: 0 to 9, A to Z, and a to z). Up to 256 bytes are allowed.

Context

Directory

Default Value

anonymous

Module

mod_ldap

AuthLDAPBindPassword

Name

AuthLDAPBindPassword

Synopsis

AuthLDAPBindPassword BindPassword

Description

When some BindDN name has been specified by the AuthLDAPBindDN directive, specify the password for the BindDN name. When making anonymous access, omit this directive.

BindPassword

Use ASCII characters (1-byte characters: 0 to 9, A to Z, and a to z). Up to 128 bytes are allowed.

Context

Directory

Default Value

None

Module

mod_ldap

AuthLDAPCertPath

Name

AuthLDAPCertPath

Synopsis

AuthLDAPCertPath operation-control-directory-name

Description

Uses the absolute path to specify the operation control directory specified when the certificate/CRL control environment was created.

Context

Directory

Default Value

None

Module

mod_ldap

AuthLDAPEnabled

Name

AuthLDAPEnabled

Synopsis

AuthLDAPEnabled on|off

Description

Specifies whether to apply LDAP authentication.

on

Applies LDAP authentication.

off

Does not apply LDAP authentication.

Context

Directory

Default Value

on

Module

mod_ldap

AuthLDAPHost

Name

AuthLDAPHost

Synopsis

AuthLDAPHost Host-name

Description

Specifies the host name including the domain name of a directory server or the IP address.

Context

Directory

Default Value

localhost

Module

mod_ldap

AuthLDAPPort

Name

AuthLDAPPort

Synopsis

AuthLDAPPort Port-number

Description

Specifies the port number of the directory server.

Context

Directory

Default Value

For not using SSL: 389

For using SSL: 636

Module

mod_ldap

AuthLDAPSecure

Name

AuthLDAPSecure

Synopsis

AuthLDAPSecure on|off

Description

Specifies whether to use SSL for the operation of the online collation function.

on

SSL is used.

off

SSL is not used.

Context

Directory

Default Value

off

Module

mod_ldap

AuthLDAPSlotPath

Name

AuthLDAPSlotPath

Synopsis

AuthLDAPSlotPath slot-information-directory-name

Description

Uses the absolute path to specify the slot information directory specified when the private-key control environment was created.

Context

Directory

Default Value

None

Module

mod_ldap

AuthLDAPTknLbl

Name

AuthLDAPTknLbl

Synopsis

AuthLDAPTknLbl token-label

Description

Specifies the token label specified when the private-key was created.

Context

Directory

Default Value

None

Module

mod_ldap

AuthLDAPTknPwd

Name

AuthLDAPTknPwd

Synopsis

AuthLDAPTknPwd user-PIN

Description

Specifies the user PIN specified when the private-key was created.

Context

Directory

Default Value

None

Module

mod_ldap

AuthName

Name

AuthName

Synopsis

AuthName 'title'

Description

Specifies the title displayed on the authentication screen in the ASCII alphanumeric characters (1 byte characters).

Context

Directory

Default value

None

AuthType

Name

AuthType

Synopsis

AuthType Basic

Description

Specifies the type of authentication 'Basic'.

Basic

Sets basic authentication (the passwords are plain text).

Context

Directory

Default Value

None

Note

Digest authentication cannot be used in Interstage HTTP Server.

<Directory>

Name

<Directory>

Synopsis

<Directory directory-path> ... </Directory>

Description

Specifies the directory section only when a directive is used within the specific directory and sub-directories of that directory.

The directory name can be specified using a relative path, wild card (? indicates a specific character, * indicates a character string), and regular expressions.

Within the specified directory, all the directives allowed in the directory context can be used.

Context

Global context, Virtual host

Default Value

None

Group

Name

Group

Synopsis

Group groupID

Description

Specifies the name of the group to use when a server process is executed.

For the group ID, the group name can be specified, or the group ID (numeric value) can be specified following a number sign (#) .

Context

Global context, Virtual host

Default Value

Solaris OE

None

Linux

#-1

Note

Group ID operates as 4294967295 when '#-1' is specified.

Initial Value

Solaris OE

```
Group nobody
```

Linux

```
Group "#-1"
```

LoadModule

Name

LoadModule

Synopsis

LoadModule module-name file-name

Description

Reads a module. Specify the file name of a module using the absolute path or the relative path from the ServerRoot directive.

Context

Global context

Default Value

None

Module

mod_so

Require

Name

Require

Synopsis

Require valid-user|user user-name|group group-name

Description

Specifies the rule to be applied for user authentication.

valid-user

Authenticates all valid users.

When the online collation function is used, users registered with the directory server are allowed.

user user-name

Authenticates users specified by *user-name*.

When the online collation function is used, the uid attribute of the user is specified as the *user-name*.

Use a space as a delimiter between user and *user-name*.

group group-name

Authenticates groups specified by *group-name*.

When the online collation function is used, the cn attribute of the group is specified as the *group-name*.

Use a space as a delimiter between group and *group-name*.

Context

Directory

Default Value

None

Examples

To authenticate a user 'taro':

```
Require user taro
```

To allow authentication of a user belonging to the group entry with the cn attribute 'ihsgroup' when the online collation function is used:

```
Require group cn=ihsgroup,ou=User,ou=interstage,o=fujitsu,dc=com
```

ServerRoot

Name

ServerRoot

Synopsis

ServerRoot directory-path

Description

Sets the root directory path in which the server lives. Relative paths for setting files are based on the directory set in this directive. Normally, conf/ and logs/ are placed under this directory as subdirectories.

Be sure to set this directive.

Context

Global context

Default Value

None

Initial Value

```
ServerRoot C:/Interstage/F3FMihs
```

```
ServerRoot /opt/FJSVihs
```

User

Name

User

Synopsis

User userID

Description

Specifies the name of the user who executes the server process.

For the user ID, the user name can be specified, or the user ID (numeric value) can be specified following a number sign (#).

Context

Global context, Virtual host

Default Value

#-1

Initial Value

```
User nobody
```

Part III

Firewall and Proxy Server

Chapter 4

HTTP Tunneling

This chapter describes HTTP Tunneling.

Note

HTTP tunneling can be used with the following products running in the Windows(R) system, Solaris OE system, or Linux system:

- Interstage Application Server Enterprise Edition
- Interstage Application Server Standard Edition

HTTP Data Communication Using HTTP Tunneling

In HTTP tunneling, data communication using the HTTP protocol can be conducted by converting data communication with the IIOP protocol used usually in CORBA applications into HTTP data. This is a useful function when you want to establish client-server linkage beyond the firewall.

HTTP Tunneling Mechanism

The following shows the HTTP tunneling mechanism:

On the client side, when HTTP tunneling is specified at start of the client application a request is sent as HTTP data during request sending from the client to the server.

On the server side, the following must be constructed: A Web server for receiving the HTTP data and HTTP gateway environment for converting HTTP data to IIOP data. A request sent to the server is converted from HTTP to IIOP data via the Web server and HTTP gateway environment. The server application (CORBA application) can receive the request as the IIOP data.

Figure 4-1 shows a processing image of the HTTP tunneling.

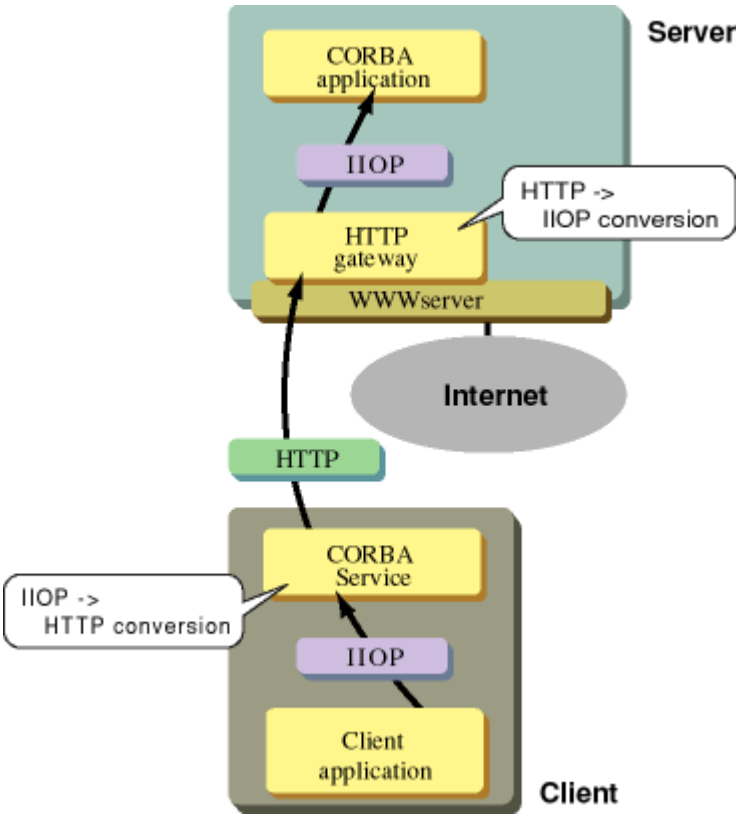


Figure 4-1 Processing Image of HTTP Tunneling

Developing the CORBA Application

When HTTP tunneling is used by a CORBA application, the ordinary CORBA application can be used intact. The application need not be recreated (including re-linkage) to use the HTTP tunneling function.

Constructing the HTTP Tunneling Environment (Constructing the HTTP Gateway Environment)

To perform data communication by the CORBA application through HTTP, the HTTP gateway environment must be constructed in the Web server.

The HTTP gateway environment enables the linkage to the following Web servers:

- Interstage HTTP Server

Windows Solaris OE

- InfoProvider Pro

Windows

- Internet Information Services

For the HTTP gateway environment, refer to HTTP Tunneling Setup.

Operating HTTP Tunneling

To use HTTP tunneling, specify parameters for using the HTTP tunneling and for specifying the HTTP gateway at start of the client application.

This enables the CORBA application to perform data communication using HTTP between the client and server.

For more information about the start parameter, refer to HTTP Tunneling Setup

HTTP Tunneling Setup

This section describes the procedure for setting the environment when using the HTTP tunneling in the CORBA application linkage.

Overview

This section describes the procedures for setting up the environment when HTTP tunneling is used.

Set up the environment for the Web server.

HTTP tunneling can be used by specifying the parameter when client applications are started.

See Setting up the Web Server Environment for details of setting up the Web server environment.

See Setting up HTTP Tunneling for details of the startup parameters.

Note

For HTTP tunneling, Interstage HTTP Server can be used as a Web server. In a Windows system, InfoProvider Pro and Microsoft Internet Information Services can be used. In a Solaris OE system, InfoProvider Pro can be used.

Refer to the following manual for details of the Web server functions, and terms related to operating procedures and commands.

Setting up the Web Server Environment

The HTTP-IIOP gateway must be established in the Web server when HTTP tunneling is used. The procedure is described below.

Establishing HTTP-IIOP Gateway

The HTTP-IIOP Gateway can be established using:

- Web Server (InfoProvider Pro), or
- Internet Information Services.

Notes

- Check the CORBA Service is operating on Web Server when the HTTP-IIOP gateway operates.

- When using the HTTP tunneling function in the IPv6 environment, the following conditions must be met.
  - The Web server needs to be operated in IPv6.
  - In the IIOP_hostname parameter in the config file, do not set an IPv6 format address or a host name that can be resolved in the IPv6 format.

(1) Using Interstage HTTP Server

Windows

Copy the following file (the installation path is the default) to the modules directory of the Interstage HTTP Server:

```
C:\Interstage\ODWIN\bin\httpgw\ODhttpAp.dll
```

Solaris OE

Copy the following file (the installation path is the default) to the libexec directory of the Interstage HTTP Server.

```
/opt/FSUNod/lib/libOMhttpAp.so
```

Linux

Copy the following file (the installation path is the default) to the libexec directory of the Interstage HTTP Server:

```
/opt/FJSVod/lib/libOMhttpAp.so
```

Open the Interstage HTTP Server environment definition file using a text editor and add the following definition to the last line.

Windows

```
LoadModule ODhttp_module modules/ODhttpAp.dll
AddModule mod_ODhttp.c
<Location /od-httpgw>
    SetHandler odhttp-handler
    Order deny,allow
    Deny from all
    Allow from all
</Location>
```

Solaris OE Linux

```
LoadModule ODhttp_module libexec/libOMhttpAp.so
AddModule mod_ODhttp.c
<Location /od-httpgw>
    SetHandler odhttp-handler
    Order deny,allow
    Deny from all
    Allow from all
</Location>
```

Notes

Solaris OE Linux

- When the Web server is Interstage HTTP Server, messages od40001 and od40002 are not output.

(2) Using InfoProvider Pro

Windows

Copy the following files to the CGI directory in the InfoProvider Pro.

Example

If the CGI directory in the InfoProvider Pro is C:\wwwhome\cgi-bin

```
copy “C:\Interstage\ODWIN\bin\httpgw\ODhttp.dll” C:\wwwhome\cgi-bin
```

Solaris OE

Copy the following files to the CGI directory in the InfoProvider Pro.

```
/opt/FSUNod/lib/libOMhttp.so
```

(3) Using Internet Information Services

Windows

Specify the C:\Interstage\ODWIN\bin\httpgw directory as the Internet Information Services virtual directory.

The procedure is as follows:

For IIS 5.0:

1. Select [Control Panel] > [Administrative Tools] > [Internet Services Manager] to start Internet Services Manager.
2. Click the icon for the local computer, and then select the target Web site.
3. Click [Operation] > [Create New] > [Virtual Directory].
4. On the virtual directory creation wizard, click [Next].
5. Enter an alias name (e.g., cgi-bin) in the Alias field, and then click [Next].
6. Specify Interstage-installation-folder\ODWIN\bin\httpgw in the Directory field, and then click [Next].
7. Check the "Execute ISAPI applications or CGIs" check box, and then click [Next].
8. Click [Finish].

For IIS 6.0:

1. Select [Control Panel] > [Administrative Tools] > [Internet Information Services (IIS) Manager] to start Internet Information Services (IIS) Manager.
2. Click the icon for the local computer, and then select the target Web site from Web Sites.
3. Click [Operation] > [Create New] > [Virtual Directory].
4. On the virtual directory creation wizard, click [Next].
5. Enter an alias name (e.g., cgi-bin) in the Alias field, and then click [Next].
6. Specify Interstage-installation-folder\ODWIN\bin\httpgw in the Path field, and then click [Next].
7. Check the "Execute ISAPI applications or CGIs" check box, and then click [Next].
8. Click [Finish].
9. Select Web Service Extensions.
10. Click [Operation] > [Add New Web Service Extension].
11. Enter an extension name in the Extension Name field.
12. Add Interstage-installation-folder\ODWIN\bin\httpgw\ODhttp.dll to Necessary Files.
13. Check the "Set extension status to permitted" check box.
14. Click [OK].

Writing HTML

To use HTTP tunneling in a Java applet, the parameter must be shown in the param tab of the applet tab in the HTML file executed by the Java applet. For details of the parameter, refer to Setting up HTTP Tunneling.

Following is an example of the HTML when a Java applet is used.

(1) For Interstage HTTP Server

```
<applet code="Sample.class" width=280 height=300>
      <param name=ORB_FJ_HTTP value=yes>
      <param name=ORB_FJ_SSL value=yes>
      <param name=ORB_FJ_HTTPGW value=http://host.com/od-httpgw>
</applet>
```

(2) For InfoProvider Pro

Windows

```
<applet code=”Sample.class” width=280 height=300>
      <param name=ORB_FJ_HTTP value=yes>
      <param name=ORB_FJ_HTTPGW value=http:/host.com/cgi-
bin/libODhttp.dll>
</applet>
```

Solaris OE

```
<applet code="Sample.class" width=280 height=300>
      <param name=ORB_FJ_HTTP value=yes>
      <param name=ORB_FJ_HTTPGW value=http:/host.com/cgi-bin/libOMhttp.so>
</applet>
```

Setting up HTTP Tunneling

In order to use HTTP tunneling, specify the parameters in Table 4-1 for the CORBA_ORB_init function called by a client application.

Application other than Java applet

Specify as a parameter when starting the application.

However, the application must be capable of passing the startup parameters to the CORBA_ORB_init function. The application presents no problem if it passes the main arguments to the CORBA_ORB_init function just as they are. Otherwise, the application logic must be modified so that the HTTP tunneling parameters are passed to the CORBA_ORB_init function.

Java applet

Use the <param> tag in the HTML file to specify the applet start time parameter.

Table 4-1 HTTP Tunneling Parameters

Parameter Name	Meaning
-ORB_FJ_HTTP	Specifies whether HTTP tunneling is used yes: HTTP tunneling is used (*1) The default, or if any value but yes is specified, is no tunneling.
-ORB_FJ_HTTPGW	Specifies the gateway that processes HTTP tunneling (*2) (1) For Interstage HTTP Server [Format for using SSL communication] https://host-name/url-name [Format for not using SSL communication] http://host-name/url-name host-name: Specifies the Web server that downloads HTML. url-name: Specifies od-httpgw. For url-name, specify the URL of the Location directive. (2) For systems other than Interstage HTTP Server http//host-name/cgi-ID/gateway-name host-name: Specify the Web server that downloads the HTML cgi-ID:

Parameter Name	Meaning
	Specify the cgi ID if Web Server is used. Windows If using Internet Information Services, specify the alias of the virtual directory. gateway-name: Windows Specify Odhttp.dll (HTTP-IIOP gateway) Solaris OE Specify libOMhttp.so (HTTP-IIOP gateway)

*1) If **yes** is specified, the HTTP tunneling function is valid if the value of argc.argv posted by the parameter in CORBA_ORB_init() is specified.

*2) The format of the host names that can be specified as arguments to be passed to "-ORB_FJ_HTTPGW" are shown below.

(1) For Interstage HTTP Server

```
http://host-name IPv4-address/URL-name
http://host-name IPv4-address:port-number/URL-name
https://host-name IPv4-address/URL-name
https://host-name IPv4-address:port-number/URL-name
```

(2) For other than Interstage HTTP Server

```
http://host-name IPv4-address/cgi-identification-name/gateway-name
http://host-name IPv4-address:port-number/cgi-identification-
name/gateway-name
http://[IPv6-address]/cgi-identification-name/gateway-name
http://[IPv6-address]:port-number/cgi-identification-name/gateway-name
https://host-name IPv4-address/cgi-identification-name/gateway-name
https://host-name IPv4-address:port-number/cgi-identification-
name/gateway-name
```

When using an address in the IPv6 format, it needs to be enclosed by square brackets ("[" and "]").

Application Other than the Java Applet

Specify the parameter in the following way when a client application (sample_c) is started:

(1) For Interstage HTTP Server

```
sample_c -ORB_FJ_HTTP yes -ORB_FJ_SSL yes
         -ORB_FJ_HTTPGW  http://host.com/od-httpgw
```

(2) For other than Interstage HTTP Server

```
sample_c –ORB_FJ_HTTP yes
         -ORB_FJ_HTTPGW http://host.com/cgi-bm/Odhttp.dll
```

```
sample_c –ORB_FJ_HTTP yes
         -ORB_FJ_HTTPGW http://host.com/cgi-bm/libOMhttp.so
```

Notes

- In client applications, use CORBA_ORB_net_disconnect() from the Fujitsu Extended Interface to disconnect the communication resource used. This is unnecessary when the Solaris OE or Linux system is used and the Web server is Interstage HTTP Server.
- When the Solaris OE or Linux system is used and the Web server is Interstage HTTP Server, no action is required because the connection with the server is released for each communication. Otherwise, client applications do not automatically disconnect the communication resource with the server at termination of the application. Thus, if the resource is not disconnected, the client application starts and stops repeatedly, the number of connections set in max_IIOP_resp_con in the config file is insufficient, and a COMM_FAILURE exception will be posted.
- Client applications created with C, C++, Java, COBOL, or OOCOBOL can be used for HTTP tunneling. They cannot be used from the OLE-CORBA gateway.

Java Applets

The HTML used for Java applets is shown below.

When the parameters are written in HTML, do not specify the hyphens in the parameter names.

(1) When Pre-installed type Java Library is Used

(1) For Interstage HTTP Server

```
<HTML>
<HEAD><!--demo.html--></HEAD>
<TITLE>Java sample Applet </TITLE>
<BODY>
<H1>Java sample Applet</H1>
```

```
<applet code="Sample.class"  width=300  height=250>
<PARAM NAME=ORB_FJ_HTTP VALUE=yes>
<PARAM NAME=ORB_FJ_SSL VALUE=yes>
<PARAM NAME=ORB_FJ_HTTPGW VALUE=http://host.com/od-httpgw>
</applet><BR>
</BODY>
</HTML>
```

(2) For other than Interstage HTTP Server

Windows

```
<HTML>
<HEAD><!—demo.html--></HEAD>
<TITLE>Java sample Applet </TITLE>
<BODY>
<HI>Java sample Applet</HI>
<applet code=”Sample.class” width=300 height=250>
<PARAM NAME=ORB_FJ_HTTP VALUE=yes>
<PARAM NAME=ORB_FJ_HTTPGW VALUE=http://host.com/cgi-bin/Odhttp.dll>
</applet><BR>
</BODY>
</HTML>
```

Solaris OE

```
<HTML>
<HEAD><!—demo.html--></HEAD>
<TITLE>Java sample Applet </TITLE>
<BODY>
<HI>Java sample Applet</HI>
<applet code=”Sample.class” width=300 height=250>
<PARAM NAME=ORB_FJ_HTTP VALUE=yes>
<PARAM NAME=ORB_FJ_HTTPGW VALUE=http://host.com/cgi-bin/libOMhttp.so>
</applet><BR>
</BODY>
</HTML>
```

(2) When Portable-ORB is used in Internet Explorer

(1) For Interstage HTTP Server

```
<HTML>
<HEAD><!--demo.html--></HEAD>
<TITLE>Java sample Applet </TITLE>
<BODY>
<H1>Java sample Applet</H1>
<applet code="Sample.class" width=300 height=250>
<PARAM NAME=cabbase VALUE=ODporb.cab,CosNaming.cab,InterfaceRep.cab>
<PARAM NAME="PORB_HOME" VALUE="PORBDIR">
<PARAM NAME=ORB_FJ_HTTP VALUE=yes>
```

```
<PARAM NAME=ORB_FJ_SSL VALUE=yes>
<PARAM NAME=ORB_FJ_HTTPGW VALUE=http://host.com/od-httpgw>
</applet><BR>
</BODY>
</HTML>
```

(2) For other than Interstage HTTP Server

Windows

```
<HTML>
<HEAD><!—demo.html--></HEAD>
<TITLE>Java sample Applet </TITLE>
<BODY>
<HI>Java sample Applet</HI>
<applet code=”Sample.class” width=300 height=250>
<PARAM NAME=cabbase VALUE=Odporb.cab.CosNaming.cab.InterfaceRep.cab>
<PARAM NAME=”PORB_HOME” VALUE=”PORBDIR”>
<PARAM NAME=ORB_FJ_HTTP VALUE=yes>>
<PARAM NAME=ORB_FJ_HTTPGW VALUE=http://host.com/cgi-bin/Odhttp.dll>
</applet><BR>
</BODY>
</HTML>
```

Solaris OE

```
<HTML>
<HEAD><!—demo.html--></HEAD>
<TITLE>Java sample Applet </TITLE>
<BODY>
<HI>Java sample Applet</HI>
<applet code=”Sample.class” width=300 height=250>
<PARAM NAME=cabbase VALUE=Odporb.cab.CosNaming.cab.InterfaceRep.cab>
<PARAM NAME=”PORB_HOME” VALUE=”PORBDIR”>
<PARAM NAME=ORB_FJ_HTTP VALUE=yes>>
<PARAM NAME=ORB_FJ_HTTPGW VALUE=http://host.com/cgi-bin/libOMhttp.so>
</applet><BR>
</BODY>
</HTML>
```

For an explanation of how to create HTML files when using Java applets, see the documents below.

- For Interstage Application Server Enterprise Edition or Interstage Application Server Standard Edition:

  Java Programming Guide in the Distributed Application Development Guide (CORBA Service Edition).

Setting to be Made When an HTTP Proxy Server is to be Used

When performing HTTP tunneling through an HTTP proxy server in the pre-installation type run-time (in an execution environment other than Portable-ORB), it is necessary to set the following in the config file of the CORBA server.

http_proxy_use: Specify whether to use HTTP tunneling (yes/no)

http_proxy: Specify the HTTP proxy server host name.

http_proxy_port: Specify the HTTP proxy server port number.

To enable the new setting of the config file, restart the CORBA service.

Example

```
http_proxy_use  = yes
http_proxy      = proxy.xxx.com
http_proxy_port = 8080
```

Note

This specification is not required if Portable-ORB is used, because the proxy server specified in the Web browser is used.

Chapter 5

HTTP Tunneling of J2EE

This chapter describes the HTTP Tunneling of J2EE.

HTTP tunneling for J2EE can be used with the following:

- J2EE application client
- IJServer (Web Applications Only)
- Java applet

HTTP tunneling for J2EE cannot be used with IJServer of any of the following types:

- Web Applications and EJB Applications run in same Java VM
- Web Applications and EJB Applications run in separate Java
- EJB Applications Only

Note

The HTTP Tunneling of J2EE can be used with the following:

- Interstage Application Server Enterprise Edition
- Interstage Application Server Standard Edition

Use of HTTP Tunneling in J2EE Application Client

To use HTTP tunneling with a J2EE application client, specify a gateway for processing HTTP tunneling using HTTPGW of the JNDI environment property.

The environment property specifies the gateway using either of the following methods.

1. The FJjndi.properties file

   The FJjndi.properties file must be placed in the following location:

   Windows

   C:\Interstage\J2EE\etc

   Solaris OE

   /etc/opt/FJSVj2ee/etc

   However, if the system name (environment property: com.fujitsu.interstage.isas.SystemName) is specified by the argument environment of new javax.naming.InitialContext(Hashtable environment) or the argument (-D) on the command line when starting an application, the FJjndi.properties file must be placed in the following location:

   /var/opt/FJSVisas/system/system_name/FJSVj2ee/etc/

   Linux

   /etc/opt/FJSVj2ee/etc

2. The argument environment of new javax.naming.InitialContext(Hashtable environment) in application.
3. The argument (-D) in command line when application starts.

If a duplicate environment property is specified, overwriting occurs in the following order (the environment property specified by step 3 takes the top priority).

1. FJjndi.properties file.
2. javax.naming.InitialContext(Hashtable environment) argument.
3. Argument (-D) on the command line when starting an application.

Windows

Environment property strings are case-sensitive.

Solaris OE Linux

Environment property and value strings are case-sensitive.

The environment property in which the gateway is specified is shown in Table 5-1.

Table 5-1 Environment Property

Environment Property	Meaning
HTTPGW	Specifies the gateway that processes HTTP tunneling. If it is omitted, tunneling is not used. (*1) (1) For Interstage HTTP Server If using encryption communication using SSL https://host-name/URL-name If not using encryption communication using SSL http://host-name/URL-name host-name: Specify the Web server that downloads the HTML URL-name: Specify od-httpgw. For URL name, specify Location directive URL Windows Solaris OE (2) For systems other than Interstage HTTP Server(Info Provider Pro) If using encryption communication using SSL https://host-name/cgi ID/gateway-name If not using encryption communication using SSL http://host-name/cgi ID/gateway-name host-name: Specify the Web server that downloads the HTML cgi-ID: Specify the cgi ID if Web Server is used Windows If using Internet Information Server, specify the alias of the virtual directory. gateway-name: Windows Specify Odhttp.dll (HTTP-IIOP gateway) Solaris OE Specify libOMhttp.so (HTTP-IIOP gateway)

*1) Solaris OE

The format of the host names that can be specified as arguments to be passed to 'HTTPGW' are shown below.

(1) For Interstage HTTP Server

```
http://ipv4address_host-name/url-name
http://ipv4address_host-name:Port_number/url-name
https://ipv4address_host-name/url-name
https://ipv4address_host-name:Port_number/url-name
```

(2) For other than Interstage HTTP Server

```
http://IPv4-address-host-name/cgi-identification-name/gateway-name
http://IPv4-address-host-name:Port_number/cgi-identification-
name/gateway-name
http://[IPv6-address]/cgi-identification-name/gateway-name
http://[IPv6-address]:Port_number/cgi-identification-name/gateway-name
https://IPv4-address-host-name/cgi-identification-name/gateway-name
https://IPv4-address-host-name:Port_number/cgi-identification-
name/gateway-name
```

When using an address in the IPv6 format, it needs to be enclosed by square brackets ("[" and "]").

In addition, in an IPv6 environment, since the SSL function cannot be used, ‘https’ cannot be specified.

An example of describing argument (-D) in the command line is shown below.

(1) For Interstage HTTP Server

```
java -DHTTPGW=http://host.com/od-httpgw SampleClient
```

(2) For other than Interstage HTTP Server

Windows

```
java -DHTTPGW=http://host.com/cgi-bin/ODhttp.dll SampleClient
```

Solaris OE

```
java -DHTTPGW=http://host.com/cgi-bin/libOMhttp.so SampleClient
```

Method for Using HTTP Tunneling with IJServer (Contains Web Applications Only)

HTTP tunneling can be used with IJServer only when IJServer is 'Web Applications Only' type. To use HTTP tunneling with web applications, HTTP tunneling parameters must be passed to the CORBA_ORB_init() function. However, when the web applications use JNDI, a gateway for processing HTTP tunneling must also be specified using HTTPGW of the JNDI environment property.

For more information about the HTTP tunneling parameters to be passed to the CORBA_ORB_init() function, refer to HTTP Tunneling in Chapter 4.

The environment property specifies the gateway using either of the following methods.

1. The FJjndi.properties file

   The FJjndi.properties file must be placed in the following location:

   Windows

   C:\Interstage\J2EE\etc

   Solaris OE

   /etc/opt/FJSVj2ee/etc

   Linux

   /etc/opt/FJSVj2ee/etc

2. Set up the JavaVM option of IJServer

   This is set up using the Interstage Management Console.

If a duplicate environment property is specified, the second method 'Set up the JavaVM option of IJServer is effective.

Environment property strings are case-sensitive.

The environment property in which the gateway is specified is shown in Table 5-2.

Table 5-2 Environment Property

Environment Property	Meaning
HTTPGW	Refer to Use of HTTP Tunneling in J2EE Application Client.

Method for Using HTTP Tunneling with Java Applets

When Java applets start, HTTP tunneling is specified with parameters of the <PARAM NAME> tags in HTML.

For information about the parameters to be specified, refer to HTTP Tunneling Parameters in Setting Up HTTP Tunneling in Chapter 4.

An example is shown below.

(1) For Interstage HTTP Server. (The items in bold text are the settings for the <PARAM NAME> parameters.)

```
<applet code="Sample.class"  width=300  height=250>
<PARAM NAME=ORB_FJ_HTTP VALUE=yes>
<PARAM NAME=ORB_FJ_SSL VALUE=yes>
<PARAM NAME=ORB_FJ_HTTPGW VALUE=https://host.com/od-httpgw>
</applet>
```

(2) For other than Interstage HTTP Server . (The items in bold text are the settings for the <PARAM NAME> parameters.)

Windows

```
<applet code=”Sample.class” width=300 height=250>
<PARAM NAME=ORB_FJ_HTTP VALUE=yes>
<PARAM NAME=ORB_FJ_SSL VALUE=yes>
<PARAM NAME=ORB_FJ_HTTPGW VALUE=https://host.com/cgi-bin/Odhttp.dll>
</applet>
```

Solaris OE

```
<applet code=”Sample.class” width=300 height=250>
<PARAM NAME=ORB_FJ_HTTP VALUE=yes>
<PARAM NAME=ORB_FJ_SSL VALUE=yes>
<PARAM NAME=ORB_FJ_HTTPGW VALUE=https://host.com/cgi-bin/libOMhttp.so>
</applet>
```

Chapter 6

Linkage of the Proxy

This chapter describes the linkage of the Proxy.

Linkage of the Proxy and SOAP Service

SOAP service can be used with the following products:

- Interstage Application Server Enterprise Edition
- Interstage Application Server Standard Edition
- Interstage Application Server Plus.

In the Interstage SOAP service, linkages between Web services through a proxy are possible.

For details, refer to the following parts in the SOAP Service User's Guide depending on the application to be created.

- RPC applications:

  Sections Basic Client Applications (Stub Method) and RPC Client Applications (DII Method) under Implementing RPC Applications.

- Messaging applications:

  Section Creating Communication Applications under Implementing Messaging Web Services.

Part IV

Authentication and Encrypted Communications through Support for SSL

This part of the manual explains how to perform encryption communication using SSL.

Encryption and signature handling require environments for controlling certificates and private keys to be configured. Interstage uses three different environments described below for those purposes. Select an environment that is suitable for the services you use and your usage situation.

For this version, use of the Interstage certificate environment is recommended.

- Interstage certificate environment (recommended)

  This environment can be shared by multiple services, and allows a unified control of certificates and private keys. You can access this environment using the Interstage Management Console.

  For details on how to configure this environment, refer to Chapter 7, Setting and Use of the Interstage Certificate Environment.

  Note: The following package is required for this environment:

  Interstage Management Console

- Certificate/key management environment configured with the SMEE command

  This environment needs to be configured separately for the respective services, using the SMEE command.

  For details on how to configure this environment, refer to Chapter 8, Setting and Use of the Certificate/Key Management Environment Using the SMEE Command.

  Note: The following package is necessary for this environment:

  Secure communication service

  (FJSVsmee package, FJSVsclr package, and FSUNssll package)

- Keystore

  The Java encryption packages JSSE (Java Secure Socket Extension) and JCE (Java Cryptography Extension) are used.

  This environment needs to be configured separately for the respective services.

  For details on how to set up this environment, refer to Chapter 14, How to prepare PKI Environment for Web Services (SOAP).

Table 7-1 shows which service can support which environment.

Table 7-1 Services and Environments

Service name	Interstage certificate environment	Certificate/key management environment	Keystore
Interstage HTTP Server	O	O	X
CORBA Service (except for the client package)	O	O	X
CORBA Service (for the client package)	X	O	X
SOAP Client	O	X	O
Servlet Service	O	X	X
EJB Server/EJB Client	X	O	X
Interstage JMS (except for the client package)	(*1)	(*1)	X
Interstage JMS (for the client package)	(*2)	(*2)	X
Event Service	(*1)	(*1)	X
Smart Repository server	O	X	X
Smart Repository client	X	O	X
Interstage Single Sign-on	(*3)	(*3)	X

[O]: Supported [X]: Not supported

*1 Uses the environment configured in the CORBA Service (except for the client package).

*2 Uses the environment configured in the CORBA Service (for the client package).

*3 Uses the environment configured in the Interstage HTTP Server.

Note

Whether the service can be used depends on the environment.

If SSL encrypted communication is to be performed in an environment configured with a version prior to V6.0, refer to the section on Authentication and Encrypted Communications through SSL Support in the previous version of the manual.

Chapter 7

Setting and Use of the Interstage Certificate Environment

This chapter explains what is required for signature and encryption processing for SSL, and explains the required processing settings.

This chapter describes a case in which the Interstage certificate environment is used.

Note

To use the Interstage certificate environment, you need to have the Interstage Management Console installed.

The Interstage certificate environment that is created here is typically used for the following services:

- Interstage HTTP Server
- CORBA Service (except client package)
- Interstage SOAP Service
- Servlet Service
- Interstage JMS

  Note: To use SSL on Interstage JMS, use the environment configured in the CORBA Service.
- Event Service

  Note: To use SSL on Event Service, use the environment configured in the CORBA Service.
- Smart Repository
- Interstage Single Sign-on

  Note: To use SSL on Interstage Single Sign-on, use the environment configured in the Interstage HTTP Server.

When you use services other than those listed above or you have not installed the Interstage Management Console, refer to Part IV of this manual, Authentication and Encrypted Communications through Support for SSL, and read the relevant section.

Certificates and Private Keys

This section explains certificates and private keys.

What are the Certificates and the Private Keys?

The CA (certification authority) certificate, site certificate, and corresponding private key are required for signature and encryption processing such as for SSL communication. A certificate revocation list (CRL) is also used to check the validity of a certificate.

A certificate and CRL that conform to X.509 or RFC2459 and that use an RSA key can be used.

- CA certificate

  This is a certificate of the CA itself that certifies the certificate issued by the CA.

  A CA may issue a certificate to a subordinate CA, in which case the certificate of the CA itself and the certificate issued to the subordinate CA are both referred to as a CA certificate. The certificate issued to the subordinate CA is specifically referred to as an intermediate CA certificate.

- Site certificate

  A site certificate is issued by a CA to certify the identity of a server, client, or service. It includes information on the user (server, client, or service) and information on the CA. The site certificate must always be used in combination with the CA certificate that issued the site certificate.

- Private key corresponding to a site certificate

  The private key forms a pair with the public key included in the site certificate.

  Note

  Losing or deleting a private-key means that the site certificate that it corresponds to is unable to be registered.

  Therefore, be sure to keep a backup of private keys.

- Certificate revocation list (CRL)

  This is a list of revoked certificates that is issued by a CA.

Table 7-2 shows the situations in which certificates including UTF-8 cannot be used. If a certificate including UTF-8 cannot be used, use a certificate that does not include UTF-8.

Table 7-2 Certificates that Can be Used

	Certificate including UTF-8	Certificate without UTF-8
SSL server authentication for Interstage SOAP Service	(*1)	O
SSL client authentication for Interstage SOAP Service	(*1)	O
SSL communication between Web server connector and Servlet container	(*1)	O
SOAP digital signature	O	O
XML encryption	O	O
Other than the above	O	O

(O: usable; X: not usable)

*1 The certificate is usable in a JDK1.4 Java execution environment but not in a JDK1.3 environment.

PKCS#12 data may be used to deliver certificates and private keys or make a backup. PKCS#12 data includes a certificate, a private key corresponding to it, and a Certification Authority certificate, all of which are password-encrypted.

In the Interstage certificate environment, you can import (register) the following types of PKCS#12 data:

- PKCS#12 data exported (extracted) with the *scsexppfx* command from the Interstage certificate environment
- PKCS#12 data exported with the *cmmkpfx* command from a certificate/key management environment created with the SMEE command

In addition, you can import PKCS#12 data exported from the Interstage certificate environment into the following environments:

- Interstage certificate environment (by means of the *scsimppfx* command)
- Certificate/key management environment created with the SMEE command (by means of the *cmentpfx* command)

For details about the certificate/key management environment, refer to Chapter 8, Setting and Use of the Certificate/Key Management Environment Using the SMEE Command.

CA (Certification Authority)

The CA (Certification Authority) is required to obtain a certificate.

The Interstage Certificate Environment supports "Secure Site" certificates issued by the VeriSign Inc. Certification Authority.

reference

Certificates issued by certification authorities other than those listed above are considered to be granted on the condition that they conform to X.509 or RFC2459. However, the operations of such certificates with Interstage Application Server as well as the acquisition processes for them have not been assured. As a result this means they are outside the scope of official support.

Configuring Environments

The Interstage Certificate Environment is an environment in which certificates, private keys, and CRLs are managed. The Interstage Certificate Environment is configured and updated using commands and referenced using the Interstage Management Console. This section describes how to configure the Interstage certificate environment.

The resources of the Interstage Certificate Environment are placed in the following directory:

Windows

```
C:\Interstage\etc\security
```

Solaris OE Linux

```
/etc/opt/FJSVisscs/security
```

You can configure the Interstage certificate environment in either of the following two ways, depending on how you obtain a certificate:

- Using a CSR (Certificate Signing Request)
- Using PKCS#12 data

The following describes the details of these two methods:

Using a CSR (Certificate Signing Request)

To ask VeriSign Inc. or another Certification Authority issuing certificates by means of a CSR to issue a certificate, or when a private Certification Authority is configured for a small-scale user using CSR, configure the environment using the following steps:

1. Solaris OE Linux

   Create an Interstage certificate environment owner group.

   Refer to Setting up Access Permissions in the Interstage Certificate Environment.

2. Configure the Interstage certificate environment.

   Refer to Configuring the Interstage Certificate Environment with CSR.

3. Make the settings for use of the certificate.

   Refer to Configuring Certificate Settings.

Using PKCS#12 Data

Use PKCS#12 data when a private Certification Authority is configured for a large-scale user and certificates are batch-issued. Configure the environment using the following steps:

1. Solaris OE Linux Create an Interstage certificate environment owner group.

   Refer to Setting up Access Permissions in the Interstage Certificate Environment

2. Configure the Interstage certificate environment.

   Refer to Configuring the Interstage Certificate Environment with PKCS#12.

3. Make the settings for use of the certificate.

   Refer to Configuring Certificate Settings.

Setting up Access Permissions in the Interstage Certificate Environment

Solaris OE Linux

Before configuring the Interstage certificate environment, you need to create owner groups allowed to access the Interstage certificate environment.

The Interstage certificate environment is configured by a superuser and accessible to effective users who belong to a specific owner group.

Effective users are assigned depending on the service. Add effective users to owner groups by service.

Although you can create or modify an owner group using an OS tool, the steps below give an example of creating an owner group using the command line.

1. Create an Interstage certificate environment owner group.

   The example below shows a command for creating a group named "iscertg".

   ```
   groupadd iscertg
   ```

2. Execute the *useradd* or *usermod* command to register an effective user in the "iscertg" group.

   The example below shows a command for adding "nobody" to "iscertg".

   ```
   usermod -G iscertg nobody
   ```

   For details about the commands, refer to the manual of the operating system you are using.

   Specify the owner group you created with the -g option of the *scsmakeenv* command when configuring the Interstage certificate environment.

Note

- Execute the commands as a superuser.
- For effective users to be registered in the Interstage certificate environment, if SSL is used on Interstage HTTP Server, you need to set users who have been specified in the User directive of the Interstage HTTP Server environment definition file (httpd.conf). The initial user value specified in the User directive is "nobody."

Windows

To give users without Administrators privileges access to the Interstage Certificate Environment, first create an Interstage Certificate Environment using the Administrators permission, then set the access permissions to allow appropriate users to access it.

This applies to the following:

- For CORBA Service, to allow users with a general permission to handle the SSL function.
- Select the Interstage Certificate Environment folder on Windows Explorer, and select a user or group to add an access permission on the "Security" tab by selecting the "Properties" menu. Set "Full Control" for the user or group that has been added.
- For details About the Interstage Certificate Environment folder, see "Setting the Environment."

Note

- If the [Security] tab is not displayed for the folder's properties on Microsoft(R) Windows(R) XP, take the following steps to display it:

1. Select [Start]-[Control Panel]-"Folder Options".
2. Click the [View] tab, remove the checkmark on "Use Simple File Sharing (Recommended)", and click "OK".

Configuring the Interstage Certificate Environment with CSR

This section describes how to configure the Interstage certificate environment using a CSR (Certificate Signing Request).

Create an Interstage certificate environment using the following steps:

1. Configure the Interstage certificate environment and create a CSR (Certificate Signing Request).
2. Request a certificate.
3. Register the certificates and CRLs.

Note that you need to set up the environment to use the certificate after configuring the Interstage certificate environment. For more details refer to Configuring Certificate Settings.

Refer to the Reference Manual (Command Edition) for details of the commands used later in this section.

Note

Windows

- Execute commands as a user with Administrators authority.

Solaris OE Linux

- Execute commands as a super user.
- Before execution, set the environment variable JAVA_HOME to the installation path for JDK or JRE.

Configuring an Interstage Certificate Environment and Creating a Certificate Signing Request (CSR)

A certificate must be obtained to perform signature and encryption processing such as for SSL. For this purpose, it is necessary to create a certificate signing request (CSR) that is the data used to request the CA to issue a certificate. When a CSR is created, an Interstage Certificate Environment is also created if it does not exist. If an Interstage Certificate Environment already exists, that certificate environment is used.

Note

Creating a CSR generates a private key in the Interstage certificate environment. To protect the private key, make a temporary backup of the Interstage certificate environment after creating the CSR and keep it until you obtain the certificate. Refer to the Operator's Guide for details of the backup procedure.

When you make no backup, you need to again create the Interstage certificate environment and create and issue a CSR, if the Interstage certificate environment is damaged.

An example of creating a CSR is shown below:

Windows

```
scsmakeenv -n SiteCert -f C:\my_folder\my_csr.txt -c
```

Solaris OE Linux

```
scsmakeenv -n SiteCert -c -f /usr/home/my_dir/my_csr.txt -g iscertg
```

Note

- The nickname specified for the -n option must be specified at registration of the site certificate. Be sure to remember the nickname.
- In an operating mode in which a service that runs as a client function obtains only server authentication, create a test certificate instead of CSR using the *scsmakeenv* command. Refer to the Reference Manual (Command Edition) for information on test certificate creation. After creating a test certificate, there is no need to perform the Requesting Certificate Issuance and Registering Certificates and CRL sections described below. Perform the sections Defining the Use of Certificates onwards.

The services listed below are concerned:

- Interstage SOAP Service
- Smart Repository (If using SSL is defined in the Replication Connection Settings in master of replication mode)
- CORBA Service (client-sided setting when SSL linkage is in use, if client authentication is not to be performed)
- Servlet Service (SSL communication setting on the Web server connector side when Web server and WorkUnit are not on the same machine)
- Interstage HTTP Server (when SSL communication with a directory server on another system is set for the online access management function)

Requesting Certificate Issuance

Request the CA to issue a certificate, and obtain a certificate.

Requesting Certificate Issuance

Send the CSR created with the *scsmakeenv* command to the CA to request certificate issuance. Follow the request procedure specified by the CA.

Obtaining a Certificate

Obtain a certificate in binary data (DER) format or Base64 encoding data (PEM) format from the CA.

A certificate in PEM format is shown below:

```
-----BEGIN CERTIFICATE-----
(Base-64-encoded certificate data is embedded.)
-----END CERTIFICATE-----
```

Follow the acquisition procedure specified by the CA.

Registering Certificates and CRL

Register the certificates and CRL obtained from the CA in the Interstage Certificate Environment.

Register the CA certificate first.

Note

After registering the obtained certificates and CRL, be sure to make a backup of the Interstage Certificate Environment. Refer to Operator's Guide for details of the backup procedure.

When you make no backup of the Interstage certificate environment, you need to again create the Interstage certificate environment and create a CSR and request certificate issuance, if the Interstage certificate environment is damaged.

Registering the CA Certificate

Register the obtained CA certificate.

An example of registration is shown below.

Windows

```
scsenter -n CA -f C:\my_folder\CA.der
```

Solaris OE Linux

```
scsenter -n CA -f /usr/home/my_dir/CA.der
```

The CA certificate can be referenced by selecting [System] > [Security] > [Certificates] > [CA certificates] from the Interstage Management Console.

Note

- The CORBA Service requires all CORBA servers and clients that use SSL to register certificates from the same CA. Refer to Chapter 10, How to Use SSL with the CORBA Service for information on how to register a certificate with the CORBA Service client package.
- All server and client systems that link to each other through communication using the Web service security function must register certificates of the same CA. Refer to Chapter 8 Setting and Use of the Certificate/Key Management Environment Using the SMEE Command for information on how to register a certificate with the Interstage SOAP Service client package.

Registering a Site Certificate

Register the issued certificate as a site certificate.

An example of registration is shown below.

Windows

```
scsenter -n SiteCert -f C:\my_folder\SiteCert.der -o
```

Solaris OE Linux

```
scsenter -n SiteCert -f /usr/home/my_dir/SiteCert.der -o
```

The site certificate can be referenced by selecting [System] > [Security] > [Certificates] > [Site certificates] from the Interstage Management Console.

Note

For the -n option, specify the same nickname as that specified when creating a CSR.

Registering the Certificate of Another Reliable Site

Register the certificate of another reliable site.

An example of registration is shown below.

Windows

```
scsenter -n OtherSiteCert -f C:\my_folder\OtherSiteCert.der -e
```

Solaris OE Linux

```
scsenter -n OtherSiteCert -f /usr/home/my_dir/OtherSiteCert.der -e
```

The certificate of another reliable site can be referenced by selecting [System] > [Security] > [Certificates] > [CA certificates] from the Interstage Management Console.

Registering a CRL

Register the obtained CRL.

An example of registration is shown below.

Windows

```
scsenter -c -f C:\my_folder\CRL.der
```

Solaris OE Linux

```
scsenter -c -f /usr/home/my_dir/CRL.der
```

Note

The SOAP client and Servlet Service (container) do not reference CRL to check for revocation.

Configuring the Interstage Certificate Environment with PKCS#12

This section describes how to configure the Interstage certificate environment using PKCS#12 data.

Create an Interstage certificate environment using the following steps:

1. Configuring the Interstage certificate environment
2. Registering PKCS#12 data, certificates, and CRLs

Note that you need to set up the environment to use the certificate after configuring the Interstage certificate environment. For details refer to Configuring Certificate Settings.

Refer to the Reference Manual (Command Edition) for details of the commands used later in this section.

Note

Windows

- Execute commands as a user with Administrators authority.

Solaris OE Linux

- Execute commands as a super user.
- Before execution, set the environment variable JAVA_HOME to the installation path for JDK or JRE.

Configuring the Interstage Certificate Environment

To set up signature handling and encryption, such as via SSL, you need to obtain a certificate and register it in the Interstage certificate environment.

This section gives an example of configuring the Interstage certificate environment.

Windows

```
scsmakeenv -e
```

Solaris OE Linux

```
scsmakeenv -e -g iscertg
```

Note

- For operation with server authentication on a service running as a client function, refer to and execute the *scsmakeenv* command to create a test certificate, not a CSR.

Registering PKCS#12 Data, Certificates, and CRLs

Register the PKCS#12 data, certificate, and CRL obtained from the Certification Authority in the Interstage certificate environment.

Note

Importing the PKCS#12 data registers the site certificate, private key corresponding to it, and the Certification Authority certificate in the Interstage certificate environment. Therefore, after importing PKCS#12 data, make a backup of the Interstage certificate environment. For details on how to make a backup, refer to the Operator's Guide.

If you don't make a backup, you need to recreate the Interstage certificate environment and import the PKCS#12 data, if the Interstage certificate environment is damaged.

Registering the CA Certificate

When a Certification Authority certificate is delivered separately from PKCS#12 data, first register the Certification Authority.

An example of registration is shown below.

Windows

```
scsenter -n CA -f C:\my_folder\CA.der
```

Solaris OE Linux

```
scsenter -n CA -f /usr/home/my_dir/CA.der
```

The CA certificate can be referenced by selecting [System] > [Security] > [Certificates] > [CA certificates] from the Interstage Management Console.

Note

- The CORBA Service requires all CORBA servers and clients that use SSL to register certificates from the same CA. Refer to Chapter 10, How to Use SSL with the CORBA Service for information on how to register a certificate with the CORBA Service client package.
- All server and client systems that link to each other through communication using the Web service security function must register certificates of the same CA. Refer to Chapter 8 Setting and Use of the Certificate/Key Management Environment Using the SMEE Command for information on how to register a certificate with the Interstage SOAP Service client package.

Importing the PKCS#12 data

Import the site certificate and private key delivered using PKCS#12 data into the Interstage certificate environment.

An example of registration is shown below.

Windows

```
scsimppfx -f C:\my_folder\MyCert.p12
```

Solaris OE Linux

```
scsimppfx -f /usr/home/my_dir/MyCert.p12
```

The site certificate can be referenced by selecting [System] > [Security] > [Certificates] > [Site certificates] from the Interstage Management Console.

Registering the Certificate of Another Reliable Site

Register the certificate of another reliable site.

An example of registration is shown below.

Windows

```
scsenter -n OtherSiteCert -f C:\my_folder\OtherSiteCert.der -e
```

Solaris OE Linux

```
scsenter -n OtherSiteCert -f /usr/home/my_dir/OtherSiteCert.der -e
```

The certificate of another reliable site can be referenced by selecting [System] > [Security] > [Certificates] > [CA certificates] from the Interstage Management Console.

Registering a CRL

Register the obtained CRL.

An example of registration is shown below.

Windows

```
scsenter -c -f C:\my_folder\CRL.der
```

Solaris OE Linux

```
scsenter -c -f /usr/home/my_dir/CRL.der
```

Note

The SOAP client and Servlet Service (container) do not reference CRL to check for revocation.

Configuring Certificate Settings

After configuring the Interstage certificate environment, you need to make the settings for users of the certificate. This section describes how to do this.

1. Defining the use of certificates.
2. Setting up various service environments.

Defining the Use of Certificates

The certificates registered in the Interstage Certificate Environment can be referenced by selecting [System] > [Security] > [Certificates] > [CA certificates], or [System] > [Security] > [Certificates] > [Site certificates] from the Interstage Management Console.

Check whether the contents of the obtained certificates are correct.

If SSL communication is to be used, an SSL definition must be created. Create an SSL definition by selecting [System] > [Security] > [SSL] > [Create a new SSL Configuration] tab from the Interstage Management Console.

Each certificate has a term of validity. Check this because system operation and functions may be stopped if the validity term of a certificate expires. It is necessary to obtain a new certificate before expiration. Refer to Certificate Management for more information.

Setting Up Various Service Environments

Environment setup (via the Interstage Management Console) is required for each service using SSL.

The option of Interstage Management Console window used for setting each service environment is shown below.

- Interstage HTTP Server

  [System] > [Services] > [Web Server] > [Web Server Settings] tab > [Detailed Settings [Show]] > [SSL Settings].

  To use SSL on the Virtual Host, you need to set up SSL on the following screens:

  – When you create a new virtual host

    [System] > [Services] > [Web Server] > [Virtual Hosts] > [Create a new Virtual Host]tab > [Detailed Settings [Show]] > [SSL Settings].

  – When you update the virtual host configuration setting

    [System] > [Services] > [Web Server] > [Virtual Hosts] > [(Virtual Host name)] > [Detailed Settings [Show]] > [SSL Settings].

- CORBA Service

  [System] > [Environment setup] tab > [Detail setting] > [CORBA service detail setting] window.

  To use SSL on a CORBA WorkUnit, you need to set up SSL on the CORBA WorkUnit setup screen. Access this screen as follows:

  [System] > [WorkUnit] > [Select WorkUnit Name] > [Deploy] tab > [Detailed Settings [View]] > [Add SSL Info to Object Reference]

- Interstage SOAP Service (*1)

  Refer to Chapter 14 How to Prepare PKI Environment for Web Services (SOAP) for environment setup details.

- Servlet Service

  Refer to "Environment Setup for Servlet Service" (Chapter Eleven, How to Use SSL with J2EE) for environment setup details.

- Interstage JMS (*1)

  [System] > [Services] > [JMS] > [EventChannels] > [Create a New Configuration] tab > [Detail Setting [Show]] > [SSL Encryption].

- Smart Repository

  For the details of the environment setting, refer to "Using SSL for Smart Repository."

- Event Service

  [System] > [Services] > [Event Service] > [EventChannels] > [Create a New Configuration] tab > [Detailed Settings [View]] > [SSL Encryption]

  Note: To use SSL in Event Service, you need to set up the SSL environment for the CORBA Service.

*1) The Interstage SOAP Service requires all linking server and client systems to register the same CA certificate.

  If the Interstage JMS or CORBA/SOAP gateway of the Interstage SOAP Service uses SSL, a CORBA Service SSL environment must be set up.

Refer to the Operator's Guide for information on starting the Interstage Management Console.

Certificate Management

After system operation begins, certificates, private keys, and CRLs must be correctly managed.

The commands shown in Table 7-3 are provided for certificate management:

Table 7-3 Certificate Management Commands

Command	Function
scsmakeenv	Configures or modifies the Interstage certificate environment. In addition, you can create a CSR or test certificate using this command
scsenter	Registers a certificate or CRL in the Interstage Certificate Environment
scslistcrl	Displays an overview of the CRL registered in the Interstage Certificate Environment
scsdelete	Deletes the site certificate with the corresponding private key, or the CA certificate from the Interstage Certificate Environment **Note**: Performing a deletion may cause the system operation or restoration to be disturbed in some cases. Read through the section "Deleting a Certificate" described later and carefully make a deletion using this command.
scsexppfx	Exports (extracts) PKCS#12 data from the Interstage certificate environment
scsimppfx	Imports (registers) PKCS#12 data to the Interstage certificate environment

These commands can be used in the situations shown below.

For command syntax and usage details, refer to the Reference Manual (Command Edition).

Note

To use the registered certificates, settings must be changed and applied using the Interstage Management Console.

Updating a Certificate (Before Expiration)

If a certificate expires, operation and function may be stopped. Before expiration, a new certificate must be obtained and registered.

After a new certificate is obtained, the current certificate is usually switched to the new one. In this case, do not delete the old certificate; leave it as is.

To switch the certificate, repeat the procedure described in Configuring Environments.

If a New Certificate and CRL are Obtained

If a new certificate is issued or a new CRL is obtained due to an increase in certificate usage after system operation begins, use the *scsenter* command to register the certificate or CRL in the Interstage Certificate Environment.

Verifying Operation using a Test Site Certificate before System Operation Begins

Before system operation begins or during application for a certificate, a test site certificate can be used to configure a system and verify operation.

Use the *scsmakeenv* command to create a test site certificate. The test site certificate is automatically registered in the Interstage Certificate Environment. There is no need to use the *scsenter* command to register the certificate.

Note

The test site certificate can be used for the following:

- Server authentication with Interstage HTTP Server
- CORBA Service with the client and server running on the same machine
- Smart Repository using SSL on Replication Connection Settings in master of replication mode.

This certificate is for testing. Do not use it for actual operation.

Deleting a Certificate

A certificate that is no longer in use can be deleted.

Note that deleting a site certificate also deletes the corresponding private key. Losing a private-key permanently disables registration of the corresponding site certificate. If the CA certificate is deleted, the CA certificate and site certificate issued by the CA can no longer be used.

Use the *scsdelete* command carefully when deleting certificates.

Retaining a certificate that is no longer in use in the environment poses no problems.

Making a PKCS#12 Data Backup and Restoring from this Backup

You can make a PKCS#12 data backup of a site certificate, private key corresponding to it, and Certification Authority certificates required for verification of the site certificate. To do this, use the *scsexppfx* command. The PKCS#12 data backup made is password-encrypted to ensure the security of the private key.

You can restore from the PKCS#12 data backup using the *scsimppfx* command. In addition, you can use the *scsimppfx* command to transfer the backup data.

However, PKCS#12 data cannot include other reliable site certificates. For information on how to make a backup of the entire Interstage certificate environment, refer to the Operator's Guide.

Chapter 8

Setting and Use of the Certificate/Key Management Environment Using the SMEE Command

This chapter describes the requirements for SSL communication and the required settings. It describes where to use a certificate/key management environment using the SMEE commands.

To use a certificate/key management environment, you need to install the secure communication services (FJSVsmee, FJSVsclr, and FSUNssll).

You can configure and use a certificate/key management environment for each of the following services:

- Interstage HTTP Server
- CORBA Service
- Servlet Service
- Interstage JMS

  Note: To use SSL with Interstage JMS, use the environment configured for the CORBA service.
- Event Service

  Note: To use SSL with Event Service, use the environment configured for the CORBA service.
- Smart Repository
- Interstage Single Sign-on

  Note: To use SSL with Interstage Single Sign-on, use the environment configured for the Interstage HTTP Server.

To use services other than those mentioned above, refer to Part IV, Authentication and Encrypted Communications through Support for SSL, and read the related chapters.

SSL Libraries Used with the Certificate/Key Management Environment

This section explains the following topics:

- Types of SMEE Libraries
- Certificate/Key Management Environment
- Environment Setting for Certificate/Key Management Environment
- Operating the Client Certificate
- Resource Registration
- Management of a Certificate/Key Management Environment

Types of SMEE Libraries

With Interstage Application Server, the following SSL libraries can be used (commands used for creating the environment vary according to the SMEE library to be used):

- SMEE2: SSL library of SMEE 2.2.x or earlier
  - Included in Interstage 1.0 and later.
  - UTF-8 certificates cannot be used.
  - To create and set up the key management environment, the *mkslt* and *mktkn* commands are used.
- SMEE3: SSL library of SMEE 3.x or later
  - Included in Interstage Application Server 4.0 or later.
  - UTF-8 certificates can be used.
  - To create and set up the key management environment, the *makeslot* and *maketoken* commands are used.

In this manual, the above libraries are referred to as SMEE2 and SMEE3.

Note

Use SMEE3 as far as possible.

Table 8-1 shows whether the above SMEE library can be used (O: usable, X: not usable).

Table 8-1 SMEE Library Usage

SSL Library	Windows Solaris OE SMEE2	SMEE3
Interstage HTTP Server	X	O
Smart Repository	X	O

SSL Library	Windows Solaris OE SMEE2	SMEE3
CORBA Service	X	O

Certificate/Key Management Environment

The following explains the certificate/key management environment, which is the operation environment when SSL (Secure Socket Layer) is used.

Certificate and Private Key

To use SSL, the CA (Certification Authorities) certificate, site certificate, and corresponding private key are required. A certificate revocation list (CRL) is also used to check the validity of a certificate.

A certificate and CRL that conform to X.509 or RFC2459 and that use an RSA key can be used.

- CA certificate

  Certificate of the CA to certify that the certificate was issued by the CA. It is also referred to as a CA certificate.

  A CA may issue a certificate to a subordinate CA, in which case the certificate of the CA itself and the certificate issued to the subordinate CA are both referred to as a CA certificate. The certificate issued to the subordinate CA is specifically referred to as an intermediate CA certificate.

- Site certificate

  Certificate issued by the CA to certify the identity of a server or client.

  This certificate contains information about the user (server, client,or service) and the CA and must always be used together with the certificate of the issuing CA.

- Private key corresponding to a site certificate

  This key is paired with a public key included in the site certificate.

  Note

  Losing a private key makes the site certificate that it corresponds to unusable. Be sure to make a back up of each private key.

- Certificate revocation list (CRL)

  This is a list of revoked certificates that is issued by a CA.

PKCS#12 data may be used to deliver certificates and private keys or make a backup. PKCS#12 data includes a certificate, a private key corresponding to it, and a Certification Authority certificate, all of which are password-encrypted.

In the Certificate/Key Management Environment, you can import (register) the following types of PKCS#12 data:

- PKCS#12 data exported (extracted) with the *scsexppfx* command from the Interstage certificate environment
- PKCS#12 data exported with the *cmmkpfx* command from a certificate/key management environment created with the SMEE command

In addition, you can import PKCS#12 data exported from the Certificate/Key Management Environment into the following environments:

- Interstage certificate environment (by means of the *scsimppfx* command)
- Certificate/key management environment created with the SMEE command (by means of the *cmentpfx* command)

CA (Certification Authority)

The CA (Certification Authority) is required to create a certificate.

The Certificate/Key Management Environment supports certificates issued by the following CAs:

- "Secure Site" certificates issued by the VeriSign Inc.

Reference

Certificates issued by other CAs(certification authorities) than those listed above are considered to be granted on the condition that they conform to X.509 or RFC2459. However, the operations of such certificates with the Interstage Application Server as well as the acquisition processes for them have not been assured. This means that they are not in the scope of official support.

Scheme of the Certificate/Key Management Environment

The certificate/key management environment is configured as shown in Figure 8-1:

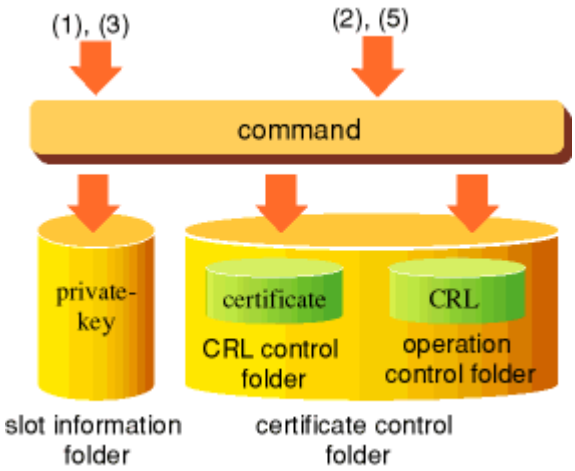


Figure 8-1 Certificate/Key Management Environment Configuration

Managing the Private Key

In key management, private keys are handled using the concept of slot and token.

The slot is an abstraction of a physical slot in which an encryption device is installed. The token is an abstraction of a physical encryption device, to be installed in the slot.

One token is allocated to one slot, but multiple private keys can be registered in one token.

Figure 8-2 shows the relationships between slot, token, and private key.

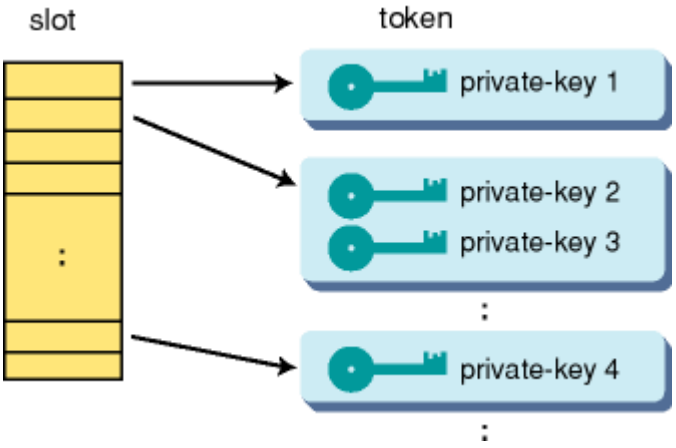


Figure 8-2 Relationship between Slot, Token and Private Key

The slot password is needed for operations processing slot information, and the SO-PIN or user PIN is needed for operations processing token information. These passwords and PINs are set when the slot is generated or when the token is generated, respectively. The SO-PIN is set and is not used in normal operation.

The user PIN refers to the information required when accessing the private key in the token (when generating a private key using the *cmmakecsr* command or registering a private key using the *cmenterkey* command). Because a user PIN exists for each token, multiple pieces of private key information can be accessed with one user PIN if multiple private keys are registered in one token.

Table 8-2 lists the relationships between password and PIN with respect to slot and token.

Table 8-2 Relationships between Password and PIN

Type	Number of pieces	Major applications
Slot-password	1 for a slot	Generating a token
SO-PIN	1 for a token	-
User PIN	1 for a token	Accessing a private key(*cmmakecsr*,*cmenterkey*)

Environment Setting for Certificate/Key Management Environment

Set up the environment according to the following procedure:

1. Create a certificate/key management environment.
   - Create management directories.
   - Create and set up a key management environment.
   - Create a certificate/CRL management environment.
2. Create a private key and acquire a certificate.
   - Create a CSR (Certificate Signing Request) (create a private key at the same time.).
   - Make a request to issue a certificate.
   - Acquire a certificate.

3. Register the certificate and CRL.
   - Register the certificate of the CA.
   - Register the site certificate.
   - Register the CRL.

For details of each command used hereafter, refer to the Reference Manual (Command Edition).

Windows

The executable files for the SMEE commands are stored in the following directory:

To set up the SSL environment, use the following commands:

- Create and set up a key management environment command for SMEE3 (makeslot maketoken)

  Under <Windows® installation drive>:\Program Files\SecurecryptoLibraryR\Program\bin
- Others

  Under <Windows® installation drive>:\Program Files\Common Files\Fujitsu Shared\F3FSSMEE

In the CORBA service, in order to execute with general user permission a CORBA application that uses the SSL linkage function, perform Steps 1 to 3 with the same user permission as the CORBA application user. If you set up an environment for certificate/key management using administrator permission, the environment will not be accessible with general user permission and this will prevent the SSL Link function from being used in the CORBA application.

If the certificate/key management environment is created with general user authority, the user who set up the environment has authority to access the environment and use the SSL linkage function. In this case, however, other general users do not have access authority for the environment. As a consequence, they cannot use SSL linkage.

To allow multiple general users to use the SSL linkage function, the certificate/key management environment access authority must be changed. Refer to “How to Use SSL with the CORBA Service” for details of access authority.

Note Solaris OE Linux

When you perform client attestation in Interstage HTTP Server, users other than super user authority need to operate Procedure 1 to 3 (since it is necessary to set up the process of a Web server by consideration on security except super user authority).

Moreover, this user and a group are set as the environmental definition file of Interstage HTTP Server. Refer to Environment Setting of the Interstage HTTP Server regarding environment setup of Interstage HTTP Server.

Creating a Certificate/Key Management Environment

Create a certificate/key management environment, which is the operation environment when using SSL.

Creating Management Directories

Four directories are required for the certificate/key management environment. Create the four directories using the commands provided by the operating system.

Example

Windows

```
mkdir  d:\sslenv\slot       # Slot information directory
mkdir  d:\sslenv\sslcert    # Operation management directory
mkdir  d:\sslenv\sslcert\cert      # Certificate management directory
mkdir  d:\sslenv\sslcert\crl# CRL management directory
```

Solaris OE Linux

```
mkdir  /export/home/slot           # Slot information directory
mkdir  /export/home/sslcert        # Operation management directory
mkdir  /export/home/sslcert\cert   # Certificate management directory
mkdir  /export/home/sslcert\crl    # CRL management directory
```

Creating and Setting Up a Key Management Environment

Create and set up the key management environment that is required for managing the private keys.

Example

If SMEE3 is used

Windows

```
makeslot -d d:\sslnewenv\slot
maketoken -d d:\sslnewenv\slot -s 1 -t Token01
```

Solaris OE Linux

```
makeslot -d /export/home/slot
maketoken -d /export/home/slot -s 1 -t Token01
```

If SMEE2 is used

Windows

```
mkslt -sd d:\sslenv\slot
#Generation and initialization of slot information directory
mktkn -sd d:\sslenv\slot -sn 1 -tl Token01
#Initial token settings
```

Solaris OE

```
mkslt -sd /export/home/slot
#Generation and initialization of slot information directory
mktkn -sd /export/home/slot -sn 1 -tl Token01
#Initial token settings
```

Creating a Certificate Management Environment

Create and set up the certificate management environment required for managing certificates and CRL.

Also, register the root certificate (CA certificate) of VeriSign Inc. using the *cmsetenv* command.

Example

Windows

```
cmmkenv d:\sslenv\sslcert –todir d:\sslenv\sslcert\cert,d:\sslenv\sslcert\crl
cmsetenv d:\sslenv\sslcert -sd d:\sslenv\slot -jc 0 -rc
C:\INTERSTAGE\IS_cert\contractcertlist
```

Solaris OE Linux

```
cmmkenv /export/home/sslcert –todir /export/home/sslcert/cert,
/export/home/sslcert/crl
cmsetenv /export/home/sslcert -sd /export/home/slot -jc 0 -rc
/etc/opt/FJSVisas/contractcertlist
```

Note

In the Interstage Application Server Web-J Edition, the storage destinations of installation certificate list files that are to be specified in the -rc option differ.

Specify an installation certificate list file using the following path:

- For using Interstage HTTP server

  /etc/opt/FJSVihs/conf/contractcertlist

Creating a Private Key and Acquiring a Certificate

Make a request to issue a certificate to the CA and acquire it.

Creating a CSR (At the Same Time Creating a Private Key)

Create a CSR to request the CA to issue a certificate.

Executing the following commands creates a private key at the same time.

Note

To secure the private key, keep a backup of the certificate/key management environment file until the certificate is obtained.

If the certificate/key management environment is damaged without a backup, the private key will be lost and it will be necessary to create the certificate/key management environment and CSR again.

Example

Windows

```
cmmakecsr -ed d:\sslenv\sslcert -sd d:\sslenv\slot -f TEXT -c jp -cn
www.infoproviderpro.com -o fujitsu -ou 4-1f -l "Shizuoka-shi" -s "Shizuoka-ken"
-kt RSA -kb 1024 –tl Token01 -of d:\sslenv\myCertRequest
ENTER  TOKEN  PASSWORD=>   *1
```

Solaris OE Linux

```
cmmakecsr -ed /export/home/sslcert -sd /export/home/slot -f TEXT -c jp -cn
www.infoproviderpro.com -o fujitsu -ou 4-1f -l "Shizuoka-shi" -s "Shizuoka-ken"
-kt RSA -kb 1024 –tl Token01 -of /export/home/myCertRequest
ENTER  TOKEN  PASSWORD=>   *1
```

*1 When these character strings appear on the screen, enter the user PIN. The entered characters are not echoed back.

Making a Request to Issue a Certificate

Send a CSR to the CA to request the issuing of a site certificate.

How to make the request depends on the CA.

Acquiring a Certificate

Acquire a certificate signed by the CA.

How to acquire a certificate depends on the CA.

Registering the Certificate and CRL

Register the obtained certificate and CRL in the certificate management environment.

After registering the certificate and CRL, make a backup of the certificate/key management environment. For details on how to make a backup, see the description of making a backup of data related to each service in Operator's Guide or "Resource Registration", "1.Search for Existing Resources (Private Key and Certificates)."

Registering the Certificate of the CA

Register the acquired certificate of the CA in the certificate management environment.

Register all certificates issued by the CA that is used for the operation (site certificates and client certificates). Use the *cmsetenv* command to register CA certificates issued by VeriSign Inc.

Register the certificates starting with the root certificate.

Example

The example below assumes the CA certificate is contained in d:\sslenv\ca-cert.der file.

```
cmentcert d:\sslenv\ca-cert.der -ed d:\sslenv\sslcert -ca -nn CACert
```

The example below assumes the CA certificate is contained in /export/home/ca-cert.der file.

```
# cmentcert /export/home/ca-cert.der -ed /export/home/sslcert -ca -nn CACert
```

Note

It is necessary to register the same issue office certificate with the CORBA Service by all the CORBA servers and CORBA clients that use SSL:

Registering the Site Certificate

Register the site certificate issued by a CA in the certificate management environment.

Example

The example below assumes the site certificate is contained in d:\sslenv\my_site_cert.der file.

```
cmentcert d:\sslenv\my_site_cert.der -ed d:\sslenv\sslcert -own -nn MySiteCert
```

Solaris OE Linux

The example below assumes the site certificate is contained in /export/home/my_site_cert.der file.

```
# cmentcert /export/home/my_site_cert.der -ed /export/home/sslcert -own -nn
MySiteCert
```

Note

For the CORBA service, if you do not authenticate clients, you do not have to register site certificates on CORBA clients.

Registering CRL

Register the CRL in the certificate management environment.

Example

Windows

The example below assumes the CRL is contained in d:\sslenv\crl.der file.

```
cmentcrl d:\sslenv\crl.der -ed d:\sslenv\sslcert
```

Solaris OE Linux

The example below assumes the CRL is contained in /entdir/crl.der file.

```
# cmentcrl /entdir/crl.der -ed /export/home/sslcert
```

Operating the Client Certificate

To perform client authentication using SSL version 3.0, the web browser needs the client certificate.

It is also required that the CA certificate that issued the client certificate be registered in the certificate/key management environment. Web Server can use client certificates issued by VeriSign Inc. only.

To perform client authentication on InfoProvider Pro, specify as follows in the SSL environment definition file:

- Specify "3" or "2 3" for the SSL version (version).
- Specify "ON" for the verification method (clcertcheck) of the client certificate.

Obtaining the Client Certificate

To obtain a client certificate, ask the Certification Authority to issue the certificate. The client certificates that Web Server can use are those issued by VeriSign Inc.

For issuance of client certificates by VeriSign Inc., refer to the companies listed in Figure 8-3.

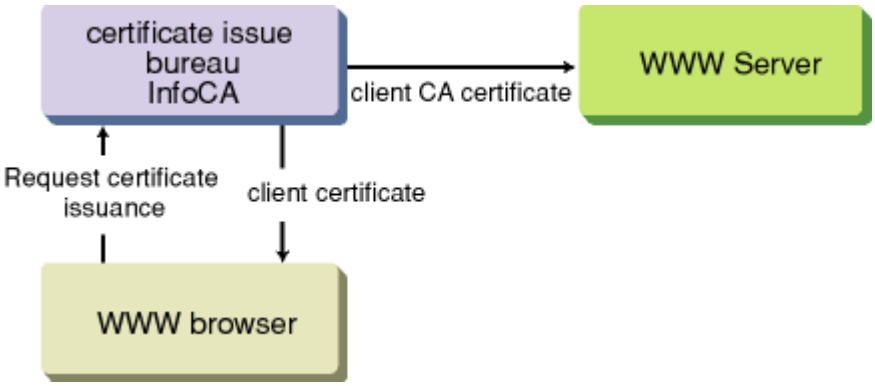


Figure 8-3 Issuance of Client Certificates

Registering the Client Certificate

Register the root certificate (CA certificate) of VeriSign Inc. when setting up the certificate/key management environment.

Example

Windows

```
cmsetenv d:\sslenv\sslcert -sd d:\sslenv\slot -jc 0 -rc
C:\INTERSTAGE\IS_cert\contractcertlist
```

Solaris OE Linux

```
cmsetenv /export/home/sslcert -sd /export/home/slot -jc 0 -rc
etc/opt/FJSVisas\contractcertlist
```

For command details, refer to the Reference Manual (Command Edition).

Resource Registration

When changing the SSL library to be used from SMEE2 to SMEE3, the existing certificate/key management environment can be used as it is.

However, to use the UTF-8 certificate, migration of the certificate/key management environment is required.

For the migration of the certificate/key management environment, create pfx data from existing resources and then register the pfx data in the new environment.

The following shows the procedure for migration:

1. Search for existing resources (private key and certificates).
2. Create a certificate/key management environment.
3. Register resources searched for in 1 in the environment created in 2.
4. Register the user PIN.

For command details, refer to the Reference Manual (Command Edition).

1. Search for Existing Resources (Private Key and Certificates).

Use the pfx data creation command to search for resources.

Example

```
cmmkpfx d:\sslenv\my_site_pfx.pfx -ed d:\sslenv\sslcert -sn 1 -nn MySiteCert
```

```
# cmmkpfx /entdir/my_site_pfx.pfx -ed /export/home/sslcert -sn 1 -nn MySiteCert
```

Client CA certificates and CRL cannot be searched for by the pfx data creation command.

If a client CA certificate or CRL is needed, re-register using the ordinary method.

When creating the pfx data, specify the nickname of the "Site certificate". The pfx data creation command reads out the site certification, its private key, the Certification Authority of the site certificate (a complete setup to the root CA certificate), and creates the pfx data.

2. Create a Certificate/Key Management Environment.

Create a certificate/key management environment.

For details, refer to Creating a Certificate/Key Management Environment.

3. Register Resources Searched For in 1. In the Environment Created in 2.

Use the pfx data registration command to register resources.

Example

The example below assumes the newly created Certificate/Key Management Environment is d:\sslnewenv\sslcert.

```
cmentpfx d:\sslenv\my_site_pfx.pfx -ed d:\sslnewenv\sslcert -sn 1 -nn
MyNewSiteCert -entca
```

Solaris OE

The example below assumes the newly created Certificate/Key Management Environment is d:\sslnewenv\sslcert.

```
# cmentpfx /entdir/my_site_pfx.pfx -ed /export/home/new/sslcert -sn 1 -nn
MyNewSiteCert -entca
```

When registering the pfx data, specify "-entca" because the CA certificate is contained in the pfx data. By doing this, the site certification, its private key, the Certification Authority of the site certificate (a complete set up to the root CA certificate) can be registered at the same time.

4. Register the User PIN.

Register the user PIN in the user PIN management file.

The following shows a registration example for InfoProvider Pro.

Example

Windows

```
ippregistupin -f d:\sslnewenv\upinfile -d d:\sslnewenv\slot
```

Solaris OE

```
ippregistupin -f /home/new/upinfile -d /home/new/slot
```

Management of a Certificate/Key Management Environment

Because each user certificate has an expiry date, re-acquisition and re-registration of certificates are needed.

The commands shown in Table 8-3 are therefore provided for managing certificates:

Table 8-3 Commands for Managing Certificates

Command	Description
cmlistcert	Displays a list of certificates registered with the certificate/key management environment.
cmdspcert	Displays contents of the specified certificate.
cmlistcrl	Displays a list of CRLs registered with the certificate/key management environment.
cmrmcert	Deletes certificates registered with the certificate/key management environment.

For information about the commands, refer to the Reference Manual (Command Edition).

Updating a Certificate (Before Expiration)

If a certificate expires, operation and function may be stopped. Before expiration, a new certificate must be obtained and registered.

After a new certificate is obtained, the current certificate is usually switched to the new one. In this case, do not delete the old certificate; leave it as is.

To switch the certificate, repeat the procedure described in Creating a Private Key and Acquiring a Certificate.

Chapter 9

How to Use SSL with Interstage HTTP Server

This chapter explains how to use the SSL for the Interstage HTTP Server.

The Interstage HTTP Server can use the following two environments for managing the certificates and private keys required for encryption and signature processing:

- Interstage certificate environment
- Certificate/key management environment configured with the SMEE command

Set either of the above SSL environments according to the operation type.

The methods for setting each SSL environment are described below.

Setting SSL for Interstage Certificate Environments

To use SSL for an Interstage certificate environment on the Interstage HTTP Server, configure the Interstage certificate environment, and then set an SSL environment using the Interstage management console. For more information, refer to Chapter 7, Setting and Use of the Interstage Certificate Environment.

Setting SSL for Certificate/Key Management Environments Configured with the SMEE Commands

On the Interstage HTTP Server, to use SSL for a certificate/key management environment configured with the SMEE commands, follow the procedure below to set an SSL environment:

1. Create a certificate/key management environment.

   For details, refer to Creating a Certificate/Key Management Environment in Environment in Chapter 8, Setting and Use of the Certificate/Key Management Environment Using the SMEE Command.

2. Create a secret key and acquire a certificate.

   For details, refer to Creating a Secret Key and Acquiring a Certificate in Chapter 8, Setting and Use of the Certificate/Key Management Environment Using the SMEE Command.

3. Register the certificate and CRL.

   For details, refer to Registering the Certificate and CRL in Environment in Chapter 8, Setting and Use of the Certificate/Key Management Environment Using the SMEE Command.

4. Register the user PIN.

5. Set the Interstage HTTP Server environment definition file.

6. Register CA certificate on the Web browser.

   For details, refer to Operating the Client Certificate in Environment in Chapter 8, Setting and Use of the Certificate/Key Management Environment Using the SMEE Command.

Note Solaris OE Linux

When performing client authentication in the Solaris OE and Linux system, a user other than the super-user authority needs to execute Steps 1 to 3. (The user other than the super-user authority set the process of Web Server for consideration on security.)
In addition, specify the user or group in the Interstage HTTP Server environment definition file in Step 5.

The following sections explain steps 4 and 5 for the Interstage HTTP Server.

Registering the User PIN

Register the user PIN in the user PIN management file.

By specifying the user PIN and user PIN management file in the *ihsregistupin* command, the user PIN is registered in the user PIN management file after encrypting it.

The following shows an example of registration.

Example Windows

When the user PIN (dialog input) is encrypted and registered to the user PIN management file "d:\ssl\upinfile".

```
ihsregistupin -f d:\ssl\upinfile -d d:\sslenv\slot
```

Example Solaris OE Linux

When the user PIN (dialog input) is encrypted and registered to the user PIN management file "/home/ssl/upinfile".

```
ihsregistupin -f /home/ssl/upinfile -d /home/sslenv/slot
```

Setting up the Environment Definition File

Make the SSL settings in the environment definition file (httpd.conf) of the Interstage HTTP Server.

In an SSL operation, by specifying the virtual host function in addition to the general operations (with or without client authentication), communication using SSL and communication without using SSL can be performed simultaneously. Examples of the environment definition files (httpd.conf) of the Interstage HTTP Server for different operations are shown below.

General Operation of SSL

Example Windows

When operating SSL using the following settings:

- Port number “443”
- Using SSL protocol version 3.0 or SSL protocol version 2.0
- Verifies a client certificate.
- Slot information directory “d:\ssl\slotdir”
- Token label “secret_key_tok”
- User PIN file “d:\ssl\upinfile”
- Operation control directory “d:\ssl\envdir”
- Nickname of the site certificate “server_cert”
- Nickname of the client CA certificate “client_cert”

```
# Add the module (Delete the comment)
AddModule mod_ihs_ssl.c

# Number of the port used for communication with a browser
Port 443
# Mail address of the server administrator
ServerAdmin webmaster@main.example.com
# Server name
```

```
ServerName main.example.com

# Using SSL
SSLExec on
# SSL protocol version
SSLVersion 2-3
# Level of client certification (Specify “none” when operating it without the
client attestation.)
SSLVerifyClient require
# Slot information directory
SSLSlotDir d:/ssl/slotdir
# Token label
SSLTokenLabel secret_key_tok
# User PIN file
SSLUserPINFile d:/ssl/upinfile
# Operation control directory
SSLEnvDir d:/ssl/envdir
# Nickname of the site certificate
SSLCertName server_cert
# Nickname of the client CA certificate
SSLClCACertName client_cert
# Method of encryption
SSLCipherSuite
RC4-MD5:RC2-MD5:EXP-RC4-MD5:RSA-RC4-MD5:RSA-RC4-SHA:RSA-EXPORT-RC4-MD5
```

Example Solaris OE Linux

When operating SSL using the following settings:

- Port number “443”
- Using SSL protocol version 3.0 or SSL protocol version 2.0
- Verifies a client certificate.
- Slot information directory “/home/ssl/slotdir”
- Token label “secret_key_tok”
- User PIN file “/home/ssl/upinfile”
- Operation control directory “/home/ssl/envdir”
- Nickname of the site certificate “server_cert”
- Nickname of the client CA certificate “client_cert”
- User of creating a certificate/key management environment “user1”
- Group of creating a certificate/key management environment “group1”

```
# Add the module (Delete the comment)
AddModule mod_ihs_ssl.c

# Number of the port used for communication with a browser
Port 443
# Mail address of the server administrator
ServerAdmin webmaster@main.example.com
```

```
# Server name
ServerName main.example.com

# User of creating a certificate/key management environment is set
User user1
# Group of creating a certificate/key management environment is set
Group group1

# Using SSL
SSLExec on
# SSL protocol version
SSLVersion 2-3
# Level of client certification (Specify “none” when operating it without the
client attestation.)
SSLVerifyClient require
# Slot information directory
SSLSlotDir /home/ssl/slotdir
# Token label
SSLTokenLabel secret_key_tok
# User PIN file
SSLUserPINFile /home/ssl/upinfile
# Operation control directory
SSLEnvDir /home/ssl/envdir
# Nickname of the site certificate
SSLCertName server_cert
# Nickname of the client CA certificate
SSLClCACertName client_cert
# Method of encryption
SSLCipherSuite
RC4-MD5:RC2-MD5:EXP-RC4-MD5:RSA-RC4-MD5:RSA-RC4-SHA:RSA-EXPORT-RC4-MD5
```

SSL Operation Using the Virtual Host Function

Example Windows

When operating SSL using the following settings:

Virtual host not using SSL:

Port number “80”, Root directory open to the public “C:\www\public”

Virtual host using SSL (without client authentication):

Port number “443”, Root directory open to the public “C:\www\secure1”

Virtual host using SSL (with client authentication):

Port number “8443”, Root directory open to the public “C:\www\secure2”

```
# Add the module (Delete the comment)
AddModule mod_ihs_ssl.c

# Number of the port used for communication with a browser
Listen 80
Listen 443
Listen 8443
```

```
# Slot information directory
SSLSlotDir d:/ssl/slotdir
# Token label
SSLTokenLabel secret_key_tok
# User PIN file
SSLUserPINFile d:/ssl/upinfile

#
# Virtual host not using SSL (Port number: 80)
#
# Sets up a virtual host
<VirtualHost 192.168.0.1:80>
  # Host name
  ServerName main.example.com
  # Root directory open to the public
  DocumentRoot C:/www/public
</VirtualHost>

#
# Virtual host using SSL (Port number:443)
#
# Sets up a virtual host
<VirtualHost 192.168.0.1:443>
  # Host name
  ServerName main.example.com
  # Root directory open to the public
  DocumentRoot C:/www/secure1
  # Using SSL
  SSLExec on
  # SSL protocol version
  SSLVersion 2
  # Operation control directory
  SSLEnvDir d:/ssl/envdir
  # Nickname of the site certificate
  SSLCertName cert_for_purchase
  # Create access log files
  CustomLog "|ihsrlog –s logs/accesslog_secure1 1 5" common
  # Create error log files
  ErrorLog "|ihsrlog -s logs/errorlog_secure1 1 5"
</VirtualHost>

#
# Virtual host using SSL (Port number:8443)
#
# Sets up a virtual host
<VirtualHost 192.168.0.1:8443>
  # Host name
  ServerName main.example.com
  # Root directory open to the public
  DocumentRoot C:/www/secure2
  # Using SSL
  SSLExec on
  # SSL protocol version
```

```
  SSLVersion 2-3
  # Level of client certification
  SSLVerifyClient require
  # Operation control directory
  SSLEnvDir d:/ssl/envdir
  # Nickname of the site certificate
  SSLCertName cert_for_manager
  # Nickname of the client CA certificate
  SSLClCACertName CACert_InfoCA
  # Method of encryption
  SSLCipherSuite
RC4-MD5:RC2-MD5:EXP-RC4-MD5:RSA-RC4-MD5:RSA-RC4-SHA:RSA-EXPORT-RC4-MD5
  # Create access log files
  CustomLog "|ihsrlog –s logs/accesslog_secure2 1 5" common
  # Create error log files
  ErrorLog "|ihsrlog -s logs/errorlog_secure2 1 5"
</VirtualHost>
```

Example Solaris OE Linux

When operating SSL using the following settings:

Virtual host not using SSL:

Port number “80”, Root directory open to the public “/home/www/public”

Virtual host using SSL (without client authentication):

Port number “443”, Root directory open to the public “/home/www/secure1”

Virtual host using SSL (with client authentication):

Port number “8443”, Root directory open to the public “/home/www/secure2”

User of creating a certificate/key management environment “user1”

Group of creating a certificate/key management environment “group1”

```
# Add the module (Delete the comment)
AddModule mod_ihs_ssl.c

# Number of the port used for communication with a browser
Listen 80
Listen 443
Listen 8443

# User of creating a certificate/key management environment is set
User user1
# Group of creating a certificate/key management environment is set
Group group1

# Slot information directory
SSLSlotDir /home/ssl/slotdir
# Token label
SSLTokenLabel secret_key_tok
# User PIN file
SSLUserPINFile /home/ssl/upinfile
```

```
#
# Virtual host not using SSL (Port number: 80)
#
# Sets up a virtual host
<VirtualHost 192.168.0.1:80>
  # Host name
  ServerName main.example.com
  # Root directory open to the public
  DocumentRoot /home/www/public
</VirtualHost>

#
# Virtual host using SSL (Port number:443)
#
# Sets up a virtual host
<VirtualHost 192.168.0.1:443>
  # Host name
  ServerName main.example.com
  # Root directory open to the public
  DocumentRoot /home/www/secure1
  # Using SSL
  SSLExec on
  # SSL protocol version
  SSLVersion 2
  # Operation control directory
  SSLEnvDir /home/ssl/envdir
  # Nickname of the site certificate
  SSLCertName cert_for_purchase
  # Create access log files
  CustomLog "|/opt/FJSVihs/bin/ihsrlog –s /opt/FJSVihs/logs/accesslog_secure1
  1 5" common
  # Create error log files
  ErrorLog "|/opt/FJSVihs/bin/ihsrlog -s /opt/FJSVihs/logs/errorlog_secure1 1
  5"
</VirtualHost>

#
# Virtual host using SSL (Port number:8443)
#
# Sets up a virtual host
<VirtualHost 192.168.0.1:8443>
  # Host name
  ServerName main.example.com
  # Root directory open to the public
  DocumentRoot /home/www/secure2
  # Using SSL
  SSLExec on
  # SSL protocol version
  SSLVersion 2-3
  # Level of client certification
  SSLVerifyClient require
  # Operation control directory
  SSLEnvDir /home/ssl/envdir
```

```
  # Nickname of the site certificate
  SSLCertName cert_for_manager
  # Nickname of the client CA certificate
  SSLClCACertName CACert_InfoCA
  # Method of encryption
  SSLCipherSuite
RC4-MD5:RC2-MD5:EXP-RC4-MD5:RSA-RC4-MD5:RSA-RC4-SHA:RSA-EXPORT-RC4-MD5
  # Create access log files
  CustomLog "|/opt/FJSVihs/bin/ihsrlog –s
/var/opt/FJSVihs/logs/accesslog_secure2 1 5" common
  # Create error log files
  ErrorLog "|/opt/FJSVihs/bin/ihsrlog -s
/var/opt/FJSVihs/logs/errorlog_secure2 1 5"
</VirtualHost>
```

Relating directives

- AddModule
- CustomLog
- DocumentRoot
- ErrorLog
- Group
- Listen
- Port
- ServerAdmin
- ServerName
- SSLCertName
- SSLCICACertName
- SSLCipherSuite
- SSLEnvDir
- SSLExec
- SSLSlotDir
- SSLTokenLabel
- SSLUserPINFile
- SSLVerifyClient
- SSLVersion
- User
- <VirtualHost>

Relating Directives

The following directives are related to the setup of the environment definition file needed to use SSL.

The description includes the following:

Name

Directive name

Synopsis

Directive format

Description

Functional overview of the directive

Context

Directive-set location indicated with one of the following keywords:

Global context

Setting used for action of the entire Web server.

Virtual host

Setting which is available in the <VirtualHost> section and used for action of the virtual host.

Directory

Setting which is available in the <Directory>, <Location>, and <Files> sections and used for action in response to a request for a specified directory, URL, or file.

Default value

Value assumed when the directive is omitted. If a directive indicated with "None" is omitted, the directive function is disabled.

Initial value

Initial directive value

Module

Name of the module that implements the directive function. A directive with no module name indication is included in the basic module.

Note

Notes on the use of the directive

Examples

Directive example (included only for a directive which requires complicated setting).

AddModule

Name

AddModule

Synopsis

AddModule module [module] ...

Description

Enables read modules or embedded modules.

Context

Global context

Default value

None

Alias

Name

Alias

Synopsis

Alias URL-path file-path|directory-path

Description

Specifies a directory to be handled as a virtual directory. Documents provided can be stored in directories other than the directory specified with the DocumentRoot directive.

The URL path can consist of up to 224 alphanumeric characters or any of the following symbols:

+, -, ., _, /.

The path must be unique. The same URL path as one specified in the ScriptAlias directive cannot be specified.

Context

Global context, Virtual host

Default value

none

Module

mod_alias

CustomLog

Name

CustomLog

Synopsis

CustomLog “|*ihsrlog-command-execution-statement*”|*log-file-name nickname* [env=[!]environment-variable]

Description

Creates access log files.

ihsrlog-command-execution-statement

Specifies the *ihsrlog* command execution statement.

log-file-name

Specifies the name of the file to which access log messages are to be output.

Specify the absolute path or the relative path from the ServerRoot directive to the log file. If the specified path does not begin with a slash "/", it is assumed to be the relative path from the ServerRoot directive.

nickname

Specifies the nickname set in the LogFormat directive.

The initial value can be any of the following nicknames:

common

- Logs access in the Common Log Format.

referer

- Logs information about follow-up of the clients.

agent

- Logs information about the Web browsers used on the clients.

combined

- Logs all information obtained with the common, referer, and agent options.

env=[!]environment-variable

Specifies that an access log message be output if the specified environment variable is already set.

If "!" is specified at the beginning of the environment variable, no access log message is output when the specified environment variable is already set.

Use the SetEnvIf directive to specify the environment variable setting conditions.

Context

Global context, Virtual host

Default value

none

Initial value

Windows

```
CustomLog "|ihsrlog -s logs/accesslog 1 5" common
```

Solaris OE Linux

```
CustomLog "|/opt/FJSVihs/bin/ihsrlog –s /var/opt/FJSVihs/logs/accesslog 1 5"
common
```

Module

mod_log_config

DocumentRoot

Name

DocumentRoot

Synopsis

DocumentRoot directory-path

Description

The directory that httpd provides the file is set.

Unless matched by a directive such as Alias, the server appends the path from the requested URL to the document root to generate the path for the document.

Be sure to set this directive.

Context

Global context, Virtual host

Default value

none

Note

Do not put the slash (/) on the end of the directory specified for this directive.

Example

Accesses "/usr/web/index.html" when specifying, "http://www.my.host.com/index.html" from a Web browser.

```
DocumentRoot  “/usr/web”
```

ErrorLog

Name

ErrorLog

Synopsis

ErrorLog “|*ihsrlog-command-execution-statement*”|*log-file-name*

Description

Creates error log files.

***ihsrlog* command execution statement**

Specifies the *ihsrlog* command execution statement.

log-file-name

Specifies the name of the file to which error log messages are to be output. For the file name, specify the absolute path or the relative path from the ServerRoot directive. If the specified path does not begin with a slash "/", it is assumed to be the relative path from the ServerRoot directive.

Context

Global context, Virtual host

Default value

Windows

logs/error.log

Solaris OE Linux

logs/error_log

Initial value

Windows

```
ErrorLog "|ihsrlog -s logs/errorlog 1 5"
```

Solaris OE Linux

```
ErrorLog "|/opt/FJSVihs/bin/ihsrlog -s /var/opt/FJSVihs/logs/errorlog 1 5"
```

Group

Name

Group

Synopsis

Solaris OE Linux

Group groupID

Description

Specifies the name of the group to use when a server process is executed.

For the group ID, the group name can be specified, or the group ID (numeric value) can be specified following a number sign (#) .

Context

Global context, Virtual host

Default value

#-1

Initial value

Solaris OE

```
Group nobody
```

Linux

```
Group #-1
```

Listen

Name

Listen

Synopsis

Listen [IP-address:]port

Description

Specifies the port numbers or IP addresses of the multiple ports that receive the connection requests from clients. Normally, ports or addresses are specified in the Port directive, but in this directive, multiple port numbers as well as IP addresses can be specified.

Context

Global context

Default value

none

Note

If possible, avoid using port numbers 1-1023 of Well Known ports when specifying a port number other than 80 (HTTP) or 443 (HTTPS). This ensures that you do not specify a port number that is already in use.

LogFormat

Name

LogFormat

Synopsis

LogFormat format [nickname]

Description

Defines the customized log format.

The format parameters that can be specified as the format are shown below.

%b

The amount of data transferred to a client (in bytes)

%h

The IP address or host name of a client

%{Foobar}i

Contents of a request header specified in Foobar

%l

Personal information of a user returned from a client

%{Cookie}n

Client IP address and a value of cookie

%r

The first line of a request

%s

A status code for a request

%t

Time and date when a request was made

%T

Duration (seconds) from the acceptance of a request from the client to completion of processing

%u

Name of a user sent from a client

%U

A path of the requested URL

For nickname, a nickname for the set format is specified.

Context

Global context, Virtual host

Default value

"%h %l %u %t \"%r\" %>s %b" clf

Initial value

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referrer
LogFormat "%{User-agent}i" agent
```

Module

mod_log_config

Port

Name

Port

Synopsis

Port *port-number*

Description

The network port number is set when the server starts. It is possible to specify it from 1 to 65535.

Context

Global context

Default value

80

Note

If possible, avoid using port numbers 1-1023 of Well Known ports when specifying a port number other than 80 (HTTP) or 443 (HTTPS). This ensures that you do not specify a port number that is already in use.

ScriptAlias

Name

ScriptAlias

Synopsis

ScriptAlias URL-path directory-path

Description

ScriptAlias sets the directory for CGI execution. Specify a virtual directory name for the URL path, and specify the absolute path of the CGI execution directory for the directory path.

The URL path can consist of up to 224 alphanumeric characters or any of the following symbols:

+, -, ., _, and /

The path must be unique. The same URL path as one specified in the Alias directive cannot be specified.

Context

Global context, Virtual host

Default value

none

Initial value

Windows

```
ScriptAlias  /cgi-bin/  "C:/Interstage/F3FMihs/cgi-bin/"
```

Solaris OE Linux

```
ScriptAlias  /cgi-bin/  "/opt/FJSVihs/cgi-bin/"
```

Module

mod_alias

ServerAdmin

Name

ServerAdmin

Synopsis

ServerAdmin *email-address*

Description

Specifies the server administrator e-mail address the server puts in error messages sent to clients.

Context

Global context, Virtual host

Default value

none

ServerName

Name

ServerName

Synopsis

ServerName *host*

Description

ServerName sets the host name or IP address of a server. The specified name is used to create a redirection URL.

Context

Global context, Virtual host

Default value

none

ServerRoot

Name

ServerRoot

Synopsis

ServerRoot directory-path

Description

Sets the root directory path in which the server lives. Relative paths for setting files are based on the directory set in this directive. Normally, conf/ and logs/ are placed under this directory as subdirectories.

Be sure to set this directive.

Context

Global context

Default value

none

Initial value

Windows

```
ServerRoot C:/Interstage/F3FMihs
```

Solaris OE Linux

```
ServerRoot /opt/FJSVihs
```

SetEnvIf

Name

SetEnvIf

Synopsis

SetEnvIf attribute attribute-value *environment-variable*[=value]

Description

Set an environment variable when attributes in a request header from a client match with specified attribute values.

As attributes, the HTTP request header information shown below must be specified. Regular expressions can be used for attribute values.

Remote_Host

Client host name

Remote_Addr

Client IP address

Remote_User

Name of the authenticated user

Request_Method

Method name

Request_Protocol

Name and version of the protocol

Request_URL

URL scheme and the portion following the host

Context

Global context, Virtual host, Directory

Default value

none

Module

mod_setenvif

SSLCertName

Name

SSLCertName

Synopsis

SSLCertName *nickname*

Description

Specifies the nickname of a site certificate registered in the certificate and CRL control environment, in up to 128 characters.

Only one SSLCertName directive can be defined for each host.

Context

Global context, Virtual host

Default value

none

Module

mod_ihs_ssl

SSLCICACertName

Name

SSLCICACertName

Synopsis

SSLCICACertName *nickname*

Description

Specifies the nickname of the CA certificate for confirming a client certificate, in up to 128 characters. This directive is used to select a specific certificate from client CA certificates registered in the operation control directory. The directive is enabled when SSL protocol version 3.0 is used.

Multiple SSLCICACertName directives can be defined for each host and each definition is enabled only for the corresponding host.

Context

Global context, Virtual host

Default value

The nicknames of all client CA certificates registered in the operation control directory are specified.

Module

mod_ihs_ssl

SSLCipherSuite

Name

SSLCipherSuite

Synopsis

SSLCipherSuite *encryption-method*

Description

Specify the methods of encryption in descending order of priority. Use colons (:) as delimiters.

When the Version 2.0 SSL protocol is used (if 2 or 2-3 is specified for the SSLVersion directive), the following values can be specified.

Table 9-1 SSLVersion Directive Values if 2 or 2-3 is specified

Value	Explanation
RC4-MD5	SSL_TXT_RC4_128_WITH_MD5 (128 bit key)
RC2-MD5	SSL_TXT_RC2_128_CBC_WITH_MD5 (128 bit key)
DES-CBC3-MD5	SSL_TXT_DES_192_EDE3_CBC_WITH_MD5 (168 bit key)
DES-CBC-MD5	SSL_TXT_DES_64_CBC_WITH_MD5 (56 bit key)
EXP-RC4-MD5	SSL_TXT_RC4_128_EXPORT40_WITH_MD5 (40 bit key)
EXP-RC2-MD5	SSL_TXT_RC2_128_CBC_EXPORT40_WITH_MD5 (40 bit key)

When the Version 3.0 SSL protocol is used (if 3 or 2-3 is specified for the SSLVersion directive), the following values can be specified.

Table 9-2 SSLVersion Directive Values if 3 or 2-3 is specified

Value	Explanation
RSA-RC4-MD5	SSL_TXT_RSA_WITH_RC4_128_MD5 (128 bit key)
RSA-RC4-SHA	SSL_TXT_RSA_WITH_RC4_128_SHA (128 bit key)
RSA-3DES-SHA	SSL_TXT_RSA_WITH_3DES_EDE_CBC_SHA (168 bit key)
RSA-DES-SHA	SSL_TXT_RSA_WITH_DES_CBC_SHA (56 bit key)
RSA-EXPORT-RC4-MD5	SSL_TXT_RSA_EXPORT_WITH_RC4_40_MD5 (40 bit key)
RSA-EXPORT-RC2-MD5	SSL_TXT_RSA_EXPORT_WITH_RC2_CBC_40_MD5 (40 bit key)
RSA-NULL-MD5	SSL_TXT_RSA_WITH_NULL_MD5
RSA-NULL-SHA	SSL_TXT_RSA_WITH_NULL_SHA

If 2-3 is specified for the SSLVersion directive, at least one value must be specified for each version.

Point

The encryption types shown in the encryption method item ("SSL_TXT_XXX") supported by Interstage Application Server are:

- Public-key encryption method: RSA
- Private-key encryption method: DES, 3DES (triple DES), RC4, RC2 (NULL means no encryption.)
- Private-key processing mode: CBC, EDE (the numerical value shows the block length.)
- Hash key: SHA, MD5

Context

Global context, Virtual host

Default value

The following values are assumed according to the specified value for the SSLVersion directive. (In the following table, each encryption method is described on a new line for clarification.)

Table 9-3 Assumed Values of SSLVersion Directive if omitted

Value of the SSLVersion directive	Default value of this directive
2	DES-CBC3-MD5: RC4-MD5: DES-CBC-MD5: EXP-RC4-MD5
3	RSA-3DES-SHA: RSA-RC4-MD5: RSA-RC4-SHA: RSA-DES-SHA: RSA-EXPORT-RC4-MD5: RSA-NULL-MD5: RSA-NULL-SHA
2-3	DES-CBC3-MD5: RC4-MD5: DES-CBC-MD5: EXP-RC4-MD5: RSA-3DES-SHA: RSA-RC4-MD5: RSA-RC4-SHA: RSA-DES-SHA: RSA-EXPORT-RC4-MD5: RSA-NULL-MD5: RSA-NULL-SHA

Module

mod_ihs_ssl

SSLEnvDir

Name

SSLEnvDir

Synopsis

SSLEnvDir *operation-control-directory-name*

Description

Specifies the operation control directory used for SSL along with the absolute path.

Only one SSLEnvDir directive can be defined for each host.

Context

Global context, Virtual host

Default value

none

Module

mod_ihs_ssl

SSLExec

Name

SSLExec

Synopsis

SSLExec [on|off]

Description

Specifies whether SSL is used.

Only one SSLExec directive can be defined for each host.

on

SSL is used.

off

SSL is not used.

Context

Global context, Virtual host

Default value

off

Module

mod_ihs_ssl

SSLSlotDir

Name

SSLSlotDir

Synopsis

SSLSlotDir *slot-information-directory*

Description

Specifies the slot information directory for the private key control environment along with the absolute path.

Only one SSLSlotDir directive can be defined for the basic area of the environment definition file (httpd.conf).

Context

Global context

Default value

none

Module

mod_ihs_ssl

SSLTokenLabel

Name

SSLTokenLabel

Synopsis

SSLTokenLabel *token-label*

Description

Specifies the token label of the token in which the private key of the server is registered, in up to 32 characters.

Only one SSLTokenLabel directive can be defined for the basic area of the environment definition file (httpd.conf).

Context

Global context

Default value

none

Module

mod_ihs_ssl

SSLUserPINFile

Name

SSLUserPINFile

Synopsis

SSLUserPINFile *user-PIN-file-name*

Description

Specifies a user PIN file along with the absolute path.

Only one SSLUserPINFile directive can be defined for the basic area of the environment definition file (httpd.conf).

For information on creating a user PIN file, see *ihsregistupin* in the Reference Manual (Command Edition).

Context

Global context

Default value

none

Module

mod_ihs_ssl

SSLVerifyClient

Name

SSLVerifyClient

Synopsis

SSLVerifyClient [none|optional|require]

Description

Specifies the level of client certification when using SSL protocol version 3.0.

Only one SSLVerifyClient directive can be defined for each host.

none

Does not verify a client certificate.

optional

Verifies a client certificate.

When a client does not provide the client certificate, the processing continues.

require

Verifies a client certificate.

When a client does not provide the client certificate, an error occurs.

When "2" is specified with the SSLVersion directive, this directive must be omitted or set to "none."

Context

Global context, Virtual host

Default value

One of the following values is specified according to the value specified with the SSLVersion directory.

Table 9-4 Assumed Values of SSLVersion Directive if omitted

Value of the SSLVersion directive	Default value of this directive
2	none
3	optional
2-3	optional

Module

mod_ihs_ssl

SSLVersion

Name

SSLVersion

Synopsis

SSLVersion [2|3|2-3]

Description

Specifies the version of SSL protocol to be used.

Only one SSLVersion directive can be defined for each host.

2

Uses SSL protocol version 2.0.

3

Uses SSL protocol version 3.0.

2-3

When SSL protocol version 3.0 can be used for communication with a client, the protocol is used. Otherwise, SSL protocol version 2.0 is used.

Context

Global context, Virtual host

Default value

2

Module

mod_ihs_ssl

User

Name

User

Synopsis

User userID

Description

Specifies the name of the user who executes the server process.

For the user ID, the user name can be specified, or the user ID (numeric value) can be specified following a number sign (#).

Context

Global context, Virtual host

Default value

#-1

Initial value

```
User nobody
```

<VirtualHost>

Name

<VirtualHost>

Synopsis

<VirtualHost> *address*[:*port*]> ... </VirtualHost>

Description

Sets up a virtual host.

The following are specified for the address.

- The IP address of the virtual host
- A fully qualified domain name for the IP address of the virtual host.

When the address is not specified, a special name "< VirtualHost _default_>" can be specified.

The port number that a virtual host uses is specified for the port. When all the port numbers are targeted, "*" is specified.

Context

Global context

Default value

none

Chapter 10

How to Use SSL with the CORBA Service

Client-server application linkage using the CORBA Service enables encrypted communication via SSL.

This chapter explains the SSL communication via the CORBA application.

In the CORBA service, the two environments listed below can be used to manage certificates and private keys required for encryption and signature processing.

- Interstage certificate environment
- Certificate/key management environment configured with the SMEE commands

Set either of the above environments according to the operation type. To use an Interstage certificate environment, refer to Chapter 7, Setting and Use of the Interstage Certificate Environment to configure the Interstage certificate environment. Then, set an SSL environment in a CORBA service using the Interstage management console.

Setting Access Permission

In an Interstage certificate environment, to run an application with permission of a common user (other than a system administrator (root)), the user must belong to an ownership group. Add the users running applications to the ownership group in the Interstage certificate environment. For more information, refer to Setting Up Access Permissions in the Interstage Certificate Environments in Chapter 7.

In a certificate/key management environment configured with the SMEE commands, to run an application with permission of a common user (other than a system administrator (root)), execute the *odsetpath* command because common user access permission must be set in a private key/certificate.

To enable a general user (a user without Administrators authority) to run an application in an Interstage certificate environment with general user authority, the access authority for the Interstage certificate environment must be changed. For details, refer to Setting up Access Permissions in the Interstage Certificate Environment in Chapter 7.

To enable a general user (a user without Administrators authority) to run an application in a certificate/key management environment configured with the SMEE command access (assuming that user is not the one that actually configured the certificate/key management environment), the following needs to be completed:

- Executing user access authority must be added to the certificate/key management environment.

Use the following procedure to add executing user access authority to the certificate/key management environment:

Use Windows Explorer to select the certificate and key management environment folders. In the [Properties] menu, [Security] tab window, add access authority by adding users and groups. Set [Full Control] for the users and groups that are added.

Note

- If the [Security] tab is not displayed for the folder's properties on Microsoft(R) Windows(R) XP, perform the following steps to display it:

1. Select [Start]-[Control Panel]-"Folder Options".
2. Click the [View] tab, remove the checkmark on "Use Simple File Sharing (Recommended)", and click "OK".

SSL Linkage of the CORBA Service

The SSL linkage function of the CORBA Service performs encrypted communication that is used for transferring data between CORBA applications by using SSL.

Mechanism of the SSL Linkage Function

If encrypted communication is set for the object reference of the server application in CORBA application linkage, encrypted communication using SSL is conducted with any client linked to the object.

Figure 10-1 shows a processing image (when an object reference is generated statically) of the SSL linkage function.

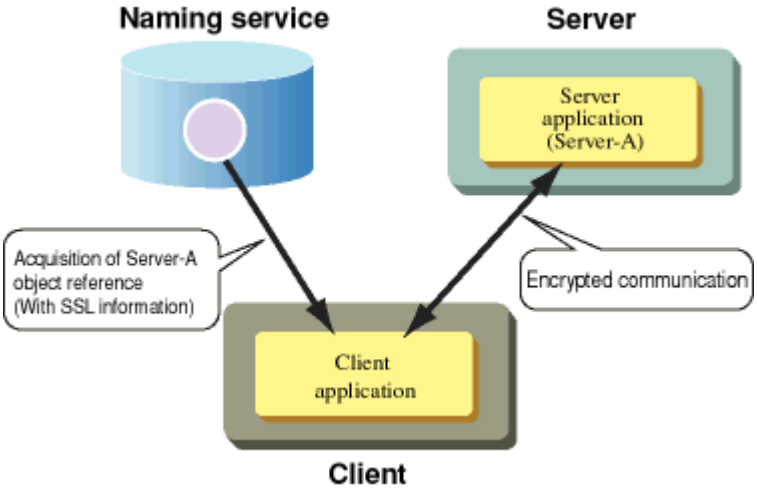


Figure 10-1 SSL Linkage Function

If SSL information is set for an object reference acquired by the client application, SSL encrypted communication is conducted to send and receive requests to the server application.

Developing the CORBA Application

To perform SSL communication using the CORBA application linkage, the ordinary CORBA application can be used as is. It is only necessary to set SSL information when the CORBA application (server application) is registered. No application needs to be recreated (including re-linkage).

Constructing SSL Linkage Environment

To perform encryption communication using SSL, the following processing must be done for the server and client: creating the certification management environment and registering the certificates.

To perform SSL communication during CORBA application operation, it is necessary to register the SSL environment in the CORBA Service and to set the SSL information for the CORBA application (server application) that performs SSL communication.

Acquiring and Registering Certificates (for both the Server and Client)

Create a private key/certificate management environment as an SSL environment, then register the CA certificate obtained from the certification authority and site certificate in the Interstage certificate environment. The same issuing office certificate must be registered for all servers and clients in which CORBA applications for SSL linkage were placed.

For an explanation of obtaining and registering certificates, refer to Chapter 7, Setting and Use of the Interstage Certificate Environment and Chapter 8, Setting and Use of the Certificate/Key Management Environment Using the SMEE Command.

Setting and Registering the SSL Environment with the CORBA Service (for both the Server and Client)

To use an Interstage certificate environment, set an SSL environment in a CORBA service using the Interstage management console.

To use a certificate/key management environment configured with the SMEE commands, first register an obtained certificate in the CORBA service using the *odsetSSL* command. Then, set the SSL linkage parameters in the operating environment file for the CORBA service (config) and incorporate SSL communication processing into the CORBA service.

Setting the SSL Information in the CORBA Application (Server Application Only)

To perform the SSL linkage using the CORBA application, the SSL information must be set in the object reference of the server application. To set the SSL information in the object reference, use the following method:

- Set the SSL information in the object reference at static generation of the object reference by the *OD_or_adm* command. (-s option)
- Specify the SSL information setting rule for the object reference generation during registration of the server application by the *OD_impl_inst* command. (ssl parameter)
  The SSL information is set according to this rule during object reference generation (both static and dynamic generations).

Operating the SSL Linkage

The application linkage that uses SSL can be performed by accessing the server application (for the object reference). SSL information is to be added to this using the *OD_or_adm* and *OD_impl_inst* commands.

SSL Linkage in the IPv6 Environment

The SSL linkage function cannot be used in the IPv6 environment.

CORBA Server Environment Setup

Configure an Interstage certificate environment, or configure a certificate/key management environment with the SMEE commands, according to the operation type. To use the Interstage certificate environment, set an SSL environment in the CORBA service using the Interstage management console.

Specifying the Addition of SSL Information at Definition of Object Reference

To perform SSL communication, execute both or either of the steps below to add SSL information to the object reference of the server application.

- Execute the -s option of the OD_or_adm command to define object reference with SSL information.
- Specify the ssl parameter in the definition file of the OD_impl_inst command, and select whether SSL information is to be added at definition of object reference.

For details on the relationships between the information specified by the OD_or_adm or OD_impl_insts Command and whether SSL communication is valid, refer to *OD_impl_inst*.

Note

Once the SSL environment has been set in the CORBA service, the *OD_or_adm* command must be used to specify the host name (-h option) and port number (-p option) if the CORBA service is not restarted.

This port number must be the same as the SSL port number for the CORBA service that has been previously set.

SSL Environment Setup in Client

To use an Interstage certificate environment, set an SSL environment using the Interstage management console. The following is the procedure for setting an environment that enables CORBA clients to use SSL in a certificate/key management environment configured with the SMEE commands.

1. Create a certificate/key management environment.

   For details, refer to Creating a Certificate/Key Management Environment in Chapter 7.

2. Create a secret key and acquire a certificate.

   For details, refer to Creating a Secret Key and Acquiring a Certificate in Chapter 7.

3. Register the certificate and CRL.

   For details, refer to Registering the Certificate and CRL in Chapter 7.

4. Define a private key/certificate in the CORBA Service.

5. Edit the config file.

The following sections explain the steps 4 and 5 for CORBA client.

Note

To set up the SSL environment, use the following commands:

To use a parameter that is too long to enter at the command prompt, include the parameter in a batch file for execution.

Table 10-1 Client SSL Setup Commands

Command	Definition
C:\Interstage\ODWIN\bin\odsetSSL	Defines private key/certificate for CORBA Service.

Defining a Private Key/Certificate in CORBA Service

To perform SSL communication via CORBA application linkage, define a private key/certificate in the CORBA Service.

Defining Private Key/Certificate

To define a private key/certificate in the CORBA Service, execute the odsetSSL command.

If the site certificate (client certificate) of the local host is used, specify the nickname of the site certificate. When the command is executed, input the user PIN set in the token.

Example

Define a private key/certificate in the CORBA Service.

```
odsetSSL  -sd C:\slot -ed C:\sslcert -tl Token01 -nn Jiro
  UserPIN:
  Re-type UserPIN:
```

Note

Do not specify a nickname (“-nn Jiro” in the above example) in operation mode without using client certificates under SSL version 3.0.

Security Attributes

For more information about the odsetSSL command, refer to *odsetSSL* in the Reference Manual (Command Edition).

The -verify option (specification of authentication when the client certificate is not present) need not be specified on the client side.

Editing config File

To embed SSL communication processing in the CORBA Service, specify "yes" in UNO_IIOP_ssl_use of the config file.

When port number 4433 (initial value) for SSL communication is used by another program, specify an unused number in the range from 1024 to 65535 in UNO_IIOP_ssl_port.

```
C:\Interstage\ODWIN\etc\config
  UNO_IIOP_ssl_use = yes
  UNO_IIOP_ssl_port = 4433
```

Note

To validate a new set value of the config file, reactivate the client application.

Environment Setup for Event Service

The Event Service can be used with the following products:

- Interstage Application Server Enterprise Edition
- Interstage Application Server Standard Edition
- Interstage Application Server Plus.

For SSL communication in the Event Service, the SSL environment of the CORBA Service must be set up. For information about the SSL environment setup of the CORBA Service, refer CORBA Server Environment Setup.

It is also necessary to set up SSL communication when setting up the Event Service environment for static generation and operation, or dynamic generation and operation. SSL communication in an event service is explained below:

For Static Generation and Operation

To create an event channel by using the *esmkchnl* command, set up SSL communication by specifying the -ssl option.

Example

For SSL communication with an event channel named CHNL1 and created in the event channel group GROUP1:

```
esmkchnl -g GROUP1 -c CHNL1 –ssl
```

For Dynamic Generation and Operation (for Environment Setting using the Interstage Integration Command)

To set up the Interstage environment, add the following definition to the Interstage operating environment definition file, and set up SSL communication.

```
Event SSL=yes
```

For Dynamic Generation and Operation (for Environment Setting using the Event Service Operation Command)

To set up an event service and an event factory using the *essetup* command, set up the SSL communication by specifying the -ssl option.

Example

To perform SSL communication via the dynamically-generated event channel when the maximum number of processes is 5 and the total number of connected suppliers and consumers on the mixed models is 100:

```
esmkchnl -g GROUP1 -c CHNL1 –ssl
```

Chapter 11

How to Use SSL with J2EE

This chapter describes how to use SSL with J2EE.

Environment Setup for Servlet Service

This section explains how to operate the Interstage Management Console.

When Performing Encrypted Communication Using SSL between the Web Browser and Web Server

Set SSL with the Web server.

On the Interstage Management Console, select [Services] > [Web Server] > [Web Server Settings] tab > [Detailed Settings[Show]], then set the following at [SSL Settings].

- Select [Enable] for [Enable SSL Encryptopn?]
- Select the SSL configuration name to be used from [SSL Configuration]

When SSL is set, the Secure attribute is automatically added to the session management cookie.

When an SSL accelerator is used the Secure attribute is not automatically added to the session management cookie, so settings must be implemented to ensure the Secure attribute is always added to the session management cookie.

On the Interstage Management Console, select [WorkUnits] > Select "IJServer WorkUnit name" > [Settings] tab > [WorkUnit settings [Show]], then set the following in [JavaVM options]:

Dcom.fujitsu.interstage.j2ee.ijserver.SessionCookieSecurity=AlwaysNeeded

When Performing Encrypted Communication Using SSL between the Web Server Connector and Servlet Container

Set the following on the Interstage Management Console:

- [Use SSL between Servlet Container and Connector?]
- [SSL Configuration to use for Servlet Container and Connector]

The setting method varies depending on the operating mode.

- When the Web server and Servlet container run on the same machine:

  Select [WorkUnits] > Select "IJServer WorkUnit name" > [Settings] tab > [Web Server Connector Settings[Show]].

- When the Web server and Servlet container run on different machines

  Set data for both the Web server connector and Servlet container:

  – Setting for Web server connector

    Select [Services] > [Web Server] > [Web Server Connector] > Select "IJServer WorkUnit name" > [Settings] tab > [Web Server Connector Settings[Show]].

  – Setting for Servlet container

    Select [WorkUnits] > Select "IJServer WorkUnit name" > [Settings] tab > [Web Server Connector Settings[Show]].

Environment Setting for EJB Service

When using SSL linkage, use the Interstage Management Console to set encrypted communication using SSL.

This setup is the same as for an application using SSL in CORBA Service. For details, refer to Chapter10, How to Use SSL with the CORBA Service.

Web-J Edition supports only the client function of the Interstage EJB service.

Steps specific to EJB Service are as follows:

Set/Unset the Encryption Communication Using the SSL Protocol

Use the Interstage Management Console to set and reset encrypted communication that uses SSL.

- Setting

  Define the EJB container of the IJServer WorkUnit that uses SSL in the window of the Interstage Management Console as follows.

  - From the Interstage Management Console, select [WorkUnit] | [IJServer name] | [Environment setup] | [EJB container setting].
  - Specify "Enable" for [Enable/disable SSL for IIOP communication].

- Unsetting

  Define the EJB container of the IJServer WorkUnit that uses SSL in the window of the Interstage Management Console as follows.

  - From the Interstage Management Console, select [WorkUnit] | [IJServer name] | [Environment setup] | [EJB container setting].
  - Specify "Disable" for [Enable/disable SSL for IIOP communication].

Note

SSL can only be used for IIOP communication when one of the following WorkUnit types applies.

- The Web application and EJB application run on different JavaVMs.
- Only an EJB application runs.

If the value for config (UNO_IIOP_ssl_port) in the CORBA Service is changed after setting up SSL communication, the EJB application must be redeployed.

When HTTP tunneling is performed using SSL communication, the start parameter must be specified when the client application starts.

For more information, refer to Chapter 5, HTTP Tunneling of J2EE.

HTTP tunneling can be used with the following products:

- Interstage Application Server Enterprise Edition
- Interstage Application Server Standard Edition

Environment Setting for Interstage JMS

Interstage JMS can be used with the following products.

- Interstage Application Server Enterprise Edition
- Interstage Application Server Standard Edition
- Interstage Application Server Plus

If SSL is to be used for Interstage JMS, settings for signature and encryption processing must be implemented such as for SSL. Refer to Chapter 8, Setting and Use of the Certificate/Key Management Environment Using the SMEE Command.

Chapter 12

Using SSL for Smart Repository

Smart Repository supports encrypted communication using SSL.

This chapter explains SSL communication for the Smart Repository.

Smart Repository can be used with the following products:

- Interstage Application Server Enterprise Edition
- Interstage Application Server Standard Edition
- Interstage Application Server Plus

SSL Communication Targets

Smart Repository supports SSL communication for the following communication paths:

Communication path	SSL communication target
Communication with command	Communication between the Smart Repository and the following commands can be encrypted: - ldapsearch command - ldapmodify command - ldapdelete command
Communication with application	Communication between the Smart Repository and client applications can be encrypted.
Communication in replication operating mode	Communication between master and slave repositories in replication operation mode can be encrypted.

Note

When the Entry Administration Tool and entry administration command (irepmodifyent) are used, SSL communication cannot be used for communication paths. Use them in a security-oriented environment by executing them on the same machine in the repository.

Client Authentication

After client authentication is performed, only the SSL client that has presented a certificate issued by a specific CA is permitted to access the SSL server. This can prevent user spoofing.

SSL linkage Environment Setup

To implement encrypted communication using SSL between a Smart Repository client and server, the SSL environment must be registered in the client and SSL information must be set for the SSL communication server.

To implement SSL communication between the master and slave in replication operation mode, the master and slave must both be set up in a certificate management environment and registered with a certificate.

Environment Setup for Using SSL between Smart Repository Client and Server

The Smart Repository client and server must both be set up in a certificate management environment for using SSL.

For the environment setup procedure, refer to "Environment Setup" - "Setting up an Environment for SSL Communication" in the Smart Repository Operator's Guide.

Environment Setup for Using SSL between Master and Slave in Smart Repository Replication Operation Mode

An Interstage certificate environment is required to use SSL between the master and slave in Smart Repository replication operation mode. After setting up an Interstage certificate environment, use the Interstage Management Console to set the SSL environment in the Smart Repository for both the master and slave servers.

For the environment setup procedure, refer to "Environment Setup" - " Setting up an Environment for SSL Communication " in the Smart Repository Operator's Guide.

Part V

Security Systems for Web Services (SOAP)

Chapter 13

Security Functions for Web Services (SOAP)

Security at the SOAP message level can be ensured by using the digital signature (SOAP digital signature) function and encryption (XML encryption) function. The reliable messaging function is used to guarantee that messages have been delivered reliably to the transmission destination and to prove that SOAP messages have been exchanged.

Digital Signature Function

The digital signature (SOAP digital signature) function is used to attach a digital signature to SOAP messages and to verify the digital signature attached to SOAP messages. By using this function, the creation or approval of the SOAP messages signed by the signer can be guaranteed. Also, by verifying the digital signature attached to the SOAP messages, it can be guaranteed that the data in the messages has not been modified.

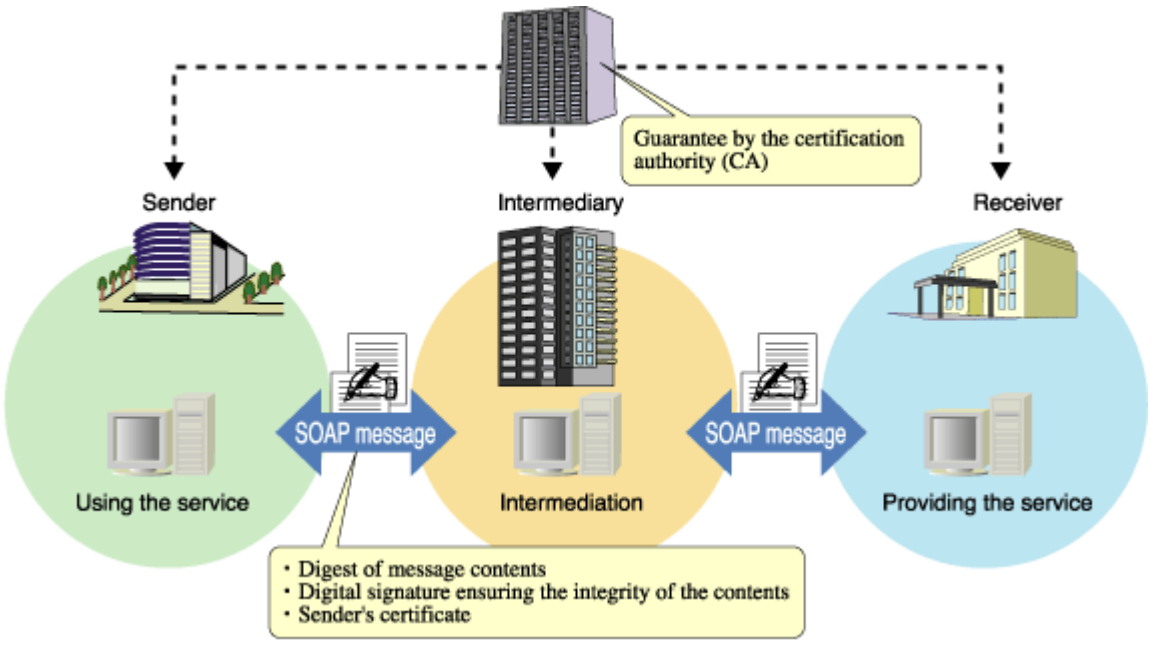


Figure 13-1 Digital Signature Function

Encryption Function of SOAP Messages

The encryption (XML encryption) function is used to encrypt communication data (XML tags in the SOAP messages and attachments) and to decrypt the encrypted communication data. By using this function, communication data can be hidden from third persons.

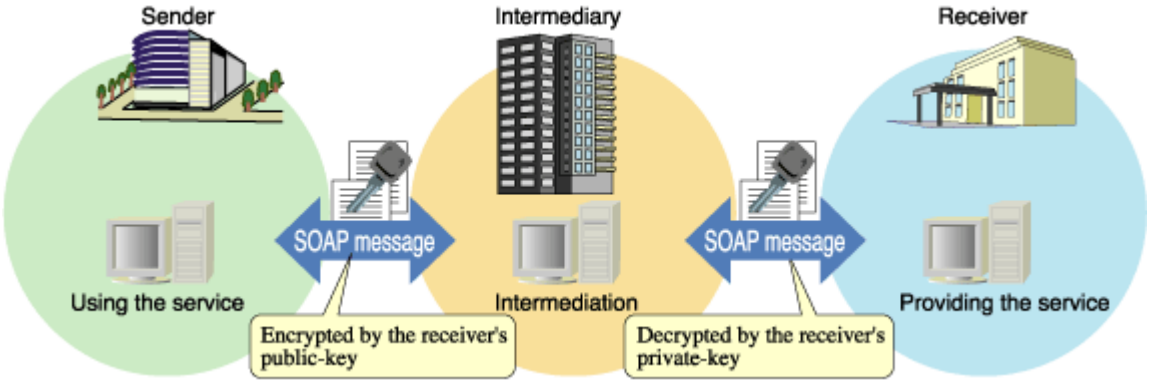


Figure 13-2 Encryption Function

Reliable Messaging Function and Non-repudiation Function

The reliable messaging function guarantees that SOAP messages to be sent will be delivered reliably to the transmission destination without duplication. By using the non-repudiation (using the signature option) function, as well, it becomes possible to exchange SOAP messages while guaranteeing reliability of both sides.

The non-repudiation function can verify later on that SOAP messages have been exchanged because SOAP messages including the transmitted SOAP digital signatures of the sender and receiver are stored automatically.

Figure 13-3 Delivery Guarantee Function and Non-repudiation Function

Attachment Function of the User ID/Password to SOAP Messages

The attachment function of the user ID/password to the SOAP messages attaches the user ID and password to the SOAP messages. By using this function, the creation or approval of SOAP messages by the owner of the user ID/password can be guaranteed (non-repudiation) without using a third-party certification authority. By combining with the XML cipher, if a mediator exists, denial by the mediator can also be prevented.

Communication via the Proxy

Client applications could exchange SOAP messages with a Web service on the Internet from within a firewall (proxy).

For more details, refer to the SOAP Service User's Guide.

Chapter 14

How to Prepare PKI Environment for Web Services (SOAP)

To allow the Web service to use SSL encrypted communication, SOAP digital signature, and XML encryption (security function), a key pair is required and a certificate management environment must be configured. This chapter explains the procedure.

Two methods are available for configuring a certificate environment:

- To configure and use a Web service security environment (Interstage certificate environment) on the server system, refer to "Configuring a Certificate Environment on the Server System."
- If the Web service security environment is already configured on the server system (old certificate environment), or to configure and use a new Web service security environment on a client system (client package certificate environment), refer to "Configuring an Old Certificate Environment or Client Certificate Environment.

Refer to Configuring an Old Certificate Environment or Client Certificate Environment for details of:

- How to execute SOAP service operation commands
- The various SOAP service commands and SOAP client applications.

Note

- To allow Web service server applications (on the server system) to perform SSL encrypted communication, configure a Web server SSL environment.
- Register the same CA certificate in every server and client system that communicate together using the security function.

Configuring a Certificate Environment on the Server System

This section explains the following aspects of configuring a certificate environment on the server system:

- Using an Interstage Certificate Environment
- Relations between Certificate Environment and Application Operation

Using an Interstage Certificate Environment

This section explains how to configure and use an Interstage certificate environment:

1. Create a private key and obtaining a certificate

   Refer to Chapter 7 Setting and Use of the Interstage Certificate Environment for details.

2. Register the certificate and CRL

   Refer to Chapter 7 Setting and Use of the Interstage Certificate Environment for details.

3. Create an SSL definition

   Create an SSL definition using the screen on the Interstage Management Console as follows:

   [System] > [Security] > [SSL] > [Create a new SSL Configuration]

4. Specify the SSL definition

   Specify the name of the SSL definition created in step 3 as a property value in the following property file:

   Windows

   %IS_HOME%\F3FMsoap\etc\config.properties

   Solaris OE Linux

   /opt/FJSVsoap/etc/config.properties

Table 14-1 Interstage Certificate Environment Properties

Property name	Value
com.fujitsu.interstage.soapx.sslname	If SSL encrypted communication, SOAP digital signature addition, or XML decryption is to be performed, specify the name of the SSL definition to be used. The site certificate can also be changed using Interstage SOAP service JavaAPI. For details, refer to "Selecting a Certificate Used for Client Authentication" in the "SOAP Service User's Guide."
com.fujitsu.interstage.soapx.websec	If SOAP digital signature addition or XML decryption is to be performed using a different SSL definition name from that used for SSL encrypted communication, specify the name of the SSL definition to be used.

Alternatively, from the Interstage Management Console, select [System] > [WorkUnits] > [MyIJServer] (IJServer WorkUnit name) > the [Settings] tab, and then [WorkUnit Settings], and specify the SSL definition name for "JavaVM Options".

Note

If a CORBA/SOAP client gateway is to be used, specify the SSL definition name in the property file.

5. Specify the SOAP digital signature/XML encryption library.

   From Interstage Management Console, select [System] > [WorkUnit] > [IJServer] > [Environment Settings] > [Common Definition]. Then, specify Fujitsu XML Processor for the XML parser to be used.

   Also, copy the following jar file:

   Windows

   <Windows® installation drive>:\Program Files\Common Files\FujitsuXML\xmltrans.jar (*1)

   Solaris OE Linux

   /opt/FJSVxmlpc/lib/xmltrans.jar(*1)

   Copy the files listed above into the following directory.

   Windows

   J2EE common directory\ijserver\IJServer workunit name\ext

   Note

   The standard of J2EE common directory is %IS_HOME%\J2EE\var\deployment.

   Solaris OE Linux

   J2EE common directory/ijserver/IJServer workunit name/ext

   Note

   The standard of J2EE common directory is /opt/FJSVj2ee/var/deployment.

6. Setting up Access Permissions. Solaris OE Linux

   Refer to Chapter 7 Setting up Access Permissions in the Interstage Certificate Environment for details.

Relations between Certificate Environment and Application Operation

Applications can operate or cannot operate depending on the combination of the following factors:

- Whether the environment is an old certificate environment (an environment configured with the soapSetSecurity or soapMngSecurity command) or an Interstage certificate environment.
- Whether the SSL definition name (property value) is specified or not.
- Whether the use of the high reliability Web service function (high reliability function) or the use of SSL encrypted communication is specified.

The table below gives a summary of application operations in the various scenarios:

Table 14-2 Summary of Application Operations

Certificate management environment	**SSL definition name specified**		**SSL definition name not specified**	
	High reliability function enabled	**SSL communication enabled**	**Interstage certificate environment only**	**Old certificate environment only**
Interstage certificate environment only	O	O	X	X
Old certificate environment only	X	X	O	O
Both Interstage certificate environment and old certificate environment	O(*1)	O(*1)	O(*2)	O(*2)

O : Can operate X : Cannot operate

(*1) Interstage certificate environment is used

(*2) Old certificate environment is used.

Configuring an Old Certificate Environment or Client Certificate Environment

Ensure that the paths (directory names and file names) required for using the Web service security functions are set in the environment variable.

Windows

Table 14-3 Environment Variable Settings

Environment variable	Description
CLASSPATH	Add the following JAR files: %IS_HOME%\F3FMsoap\lib\issoapsec.jar(*1) %IS_HOME%\J2EE\lib\jcert.jar (*2) %IS_HOME%\J2EE\lib\jnet.jar (*2) %IS_HOME%\J2EE\lib\jsse.jar (*2) %IS_HOME%\J2EE\lib\jce1_2_2.jar %IS_HOME%\J2EE\lib\local_policy.jar %IS_HOME%\J2EE\lib\sunjce_provider.jar %IS_HOME%\J2EE\lib\US_export_policy.jar If a SOAP digital signature or XML encryption is to be used, specify the following JAR file as well: <Windows® installation drive>:\Program Files\Common Files\FujitsuXML\xmlpro.jar(*1) <Windows® installation drive>:\Program Files\Common Files\FujitsuXML\xmltrans.jar (*1) Specify one of the following additional jar files when using the user authentication function in a server system environment: [JDK1.3.1] %IS_HOME%\F3FMsso\ssoatzag\lib\isssomod.jar [JDK1.4] %IS_HOME%\F3FMsso\ssoatzag\lib\isssomod14.jar

(*1) Set the environment variable CLASSPATH prior to xmldsig.jar, xmlpro.jar, or xmltrans.jar included in another product and another function.

(*2) Set the environment variable CLASSPATH prior to %IS_HOME%\J2EE\lib\isj2ee.jar.

Solaris OE Linux

Table 14-4 Environment Variable Settings

Environment variable	Description
CLASSPATH	Add the following JAR files: /opt/FJSVsoap/lib/issoapsec.jar(*1) /opt/FJSVj2ee/lib/jcert.jar (*2) /opt/FJSVj2ee/lib/jnet.jar (*2) /opt/FJSVj2ee/lib/jsse.jar (*2) /opt/FJSVj2ee/lib/jce1_2_2.jar /opt/FJSVj2ee/lib/local_policy.jar /opt/FJSVj2ee/lib/sunjce_provider.jar /opt/FJSVj2ee/lib/US_export_policy.jar /opt/FJSVj2ee/lib/isj2ee.jar If a SOAP digital signature or XML encryption is to be used, specify the following JAR file as well: /opt/FJSVxmlpc/lib/xmlpro.jar (*1) /opt/FJSVxmlpc/lib/xmltrans.jar (*1) Specify one of the following additional jar files when using the user authentication function in a server system environment: [JDK1.3.1] IS_HOME%\F3FMsso\ssoatzag\lib\isssomod.jar [JDK1.4] %IS_HOME%\F3FMsso\ssoatzag\lib\isssomod14.jar
JAVA_HOME	Set the following directory: - Installation directory of JDK
PATH	Set the following directories: $JAVA_HOME/bin /opt/FJSVsoap/bin
LD_LIBRARY_PATH	Linux Set the following directories: /opt/FJSVsoap/tools

(*1) Set the environment variable CLASSPATH prior to xmldsig.jar, xmlpro.jar, or xmltrans.jar included in another product and another function.

(*2) Set the environment variable CLASSPATH prior to /opt/FJSVj2ee/lib/isj2ee.jar.

Constructing a Key Pair/Certificate Management Environment

If the security function is to be used in the Web service, how to construct a key pair/certificate management environment depends on the functions to be used and the environment (whether a server system or a client system).

This section explains how to construct and manage a key pair/certificate management environment using the *soapSetSecurity* and *soapMngSecurity* commands.

Constructing a Key Pair/Certificate Management Environment

Construct the following environment and then acquire a certification authority certificate and a site certificate from the certification authority for registration.

- Web service security environment information file
- File used to register and manage the key pairs (public-key and private-key), certification authority certificates and site certificates (hereafter called "the certificate management file")

Figure 14-1 shows the procedure for constructing a key pair/certificate management environment used by the security function.

Create and set key pair and certificate management environment
Acquire certificate from certification authority
Register certificate

Figure 14-1 Constructing a Key Pair/Certificate Management Environment

In the following cases the creation of a key pair and the acquisition of a site certificate from the certification authority is necessary:

- To generate the SOAP digital signature.
- Decryption using the XML encryption.
- Client authentication in SSL-encrypted communication.

Note

Refer to Environment Construction when a Private-key is needed.

In the following cases the creation of a key pair and the acquisition of a site certificate from the certification authority can be omitted:

- If the SOAP digital signature is verified.
- Data is encrypted using XML encryption.
- The client is not authenticated in SSL-encrypted communication.

Note

Refer to Environment Construction when a Private-key is not Needed.

The following encryption methods can be used in SSL-encrypted communication.

Table 14-5 Encryption Methods that can be Used

encryption method
SSL_DHE_DSS_WITH_DES_CBC_SHA
SSL_DHE_DSS_WITH_3DES_EDE_CBC_SHA
SSL_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
SSL_RSA_WITH_RC4_128_MD5
SSL_RSA_WITH_RC4_128_SHA
SSL_RSA_WITH_DES_CBC_SHA
SSL_RSA_WITH_3DES_EDE_CBC_SHA
SSL_RSA_EXPORT_WITH_RC4_40_MD5

Environment Construction when a Private-key is needed

To use the following functions in the Web service, the creation of a private-key and the acquisition of a site certificate are needed.

- If the client should be authenticated in SSL-encrypted communication.
- SOAP digital signature generation.
- Decryption using the XML encryption.

Creating and Setting a Key Pair/Certificate Management Environment

Use the XML encryption to create a key pair/certificate management environment required for the SSL client authentication, the SOAP digital signature generation or decryption.

- Create a directory in which the file (the certificate management file) used to create and manage key pairs and to register and manage certificates is to be placed.
- Create a Web service security environment information file and the certificate management file using the *soapSetSecurity* command (key pair/certificate management environment creation). How to use *soapSetSecurity* depends on the certification authority to which the application for a certificate is made.

The following shows some examples of using the *soapSetSecurity* command for each certification authority to be used.

Example

If SystemWalker/PkiMGR is the certification authority.

Example 1. Windows

Create a Web service security environment information file and the certificate management file in the certificate management file creation directory by specifying the alias (taro) and the password for the certificate management file access (Interstage). The installation directory of Interstage is assumed to be "C:\Interstage".

```
soapSetSecurity  -f C:\Interstage\F3FMsoap\etc  -p Interstage  -alias taro
```

Example 2. Solaris OE Linux

Create a Web service security environment information file and the certificate management file in the certificate management file creation directory by specifying the alias (taro) and the password for the certificate management file access (Interstage).

```
soapSetSecurity  -f /etc/opt/FJSVsoap/etc  -p Interstage  -alias taro
```

If VeriSign Japan K.K or Japan Certification Services Inc. is the certification authority.

Example 3. Windows

To register the root certificates issued by VeriSign Japan K.K or Japan Certification Services Inc., specify the root certificate list file (C:\Interstage\IS_cert\contractcertlist) in the -rc option of the *soapSetSecurity* command.

```
soapSetSecurity  -f C:\Interstage\F3FMsoap\etc  -p Interstage  -alias taro
               -rc C:\Interstage\IS_cert\contractcertlist
```

Example 4. Solaris OE Linux

To register the root certificates issued by VeriSign Japan K.K or Japan Certification Services Inc., specify the root certificate list file (/etc/opt/FJSVisas/contractcertlist) in the -rc option of the *soapSetSecurity* command.

```
soapSetSecurity  -f /etc/opt/FJSVsoap/etc  -p Interstage  -alias taro
               -rc C:\Interstage\IS_cert\contractcertlist
```

The following certificates are stored in the certificate management file as the root certificates:

- Root certificates issued by VeriSign Japan K.K.
  - VeriSign/RSA Secure Server CA
  - VeriSign Class 1 Public Primary Certification Authority
  - VeriSign Class 2 Public Primary Certification Authority
  - VeriSign Class 3 Public Primary Certification Authority
  - VeriSign Class 1 Public Primary Certification Authority - G2
  - VeriSign Class 2 Public Primary Certification Authority - G2
  - VeriSign Class 3 Public Primary Certification Authority - G2
  - VeriSign Class 4 Public Primary Certification Authority - G2

- Root certificates issued by Japan Certification Services Inc.
  - SecureSign RootCA1
  - SecureSign RootCA2
  - SecureSign RootCA3

Acquiring Certificates From The Certification Authority

Creating a Certificate Signing Request

Create a certificate signing request (CSR) to make a request to the certification authority to issue a site certificate. A CSR corresponding to a private-key can be created, after creating a public-key/private-key pair using the *soapSetSecurity* (key pair/certificate management environment creation), by using the *soapMngSecurity* (certificate management) command.

Example

```
soapMngSecurity  -certreq  -f certificate_application_storage_file_name -p
                 Interstage  -alias taro
```

The password and alias specified as the options must be the same as those specified when a key pair/certificate management environment is created using the *soapSetSecurity* command.

Requesting to Issue a Certificate

Send a certificate-signing request to the certification authority to make a request to issue a certification authority certificate and a site certificate. Follow the procedure of each certification authority to make a request for a certificate.

Acquiring a Certificate

Acquire a certificate signed by the certification authority. Follow the procedure of each certification authority to acquire a certificate.

Note

If a certificate is to be used for the SOAP digital signature, the digital signature must be specified as the usage of the public-key contained in the site certificate to be acquired from the certification authority.

Registering Certificates

Register the site certificates and certification authority certificates with the certificate management file. Certificates in the format containing the site certificates and certification authority certificates (such as PKCS#7), or containing only one site certificate or certification authority certificate can be registered using the *soapMngSecurity* (certificate management) command.

Notes

- The same certification authority certificates must be registered on all servers and clients linked for communication using the security function of the Web service.
- Before registering a site certificate, the certificate of the certification authority must be registered using an alias. If the certification authority is an intermediate certification authority, registration of the certificate must start from the root certification authority.

Example

Register the site certificate and certification authority certificate with the certificate management file by specifying them.

```
soapMngSecurity  -import  -f
                certification_authority_certificate_storage_file_name
                -p Interstage  -alias cacert
soapMngSecurity  -import  -f  site_certificate_storage_file_name  -p
Interstage
               -alias taro -own
```

Environment Construction when a Private-key is not Needed

To use the following functions in the Web service, it is not necessary to create a private-key and acquire a site certificate from the certification authority. However, it is necessary to register the certification authority certificates and site certificates of the communication parties that use the security function.

- If the client is not to be authenticated in SSL communication.
- SOAP digital signature verification.
- Encryption using the XML encryption.

Creating and Setting a Certificate Management Environment

Use the XML encryption to create a certificate management environment required for the SSL-encrypted communication, SOAP digital signature verification or encryption.

- Create a directory in which the certificate management file (used to register and manage certificates) will be placed.
- Create a Web service security environment information file (the certificate management file) using the *soapSetSecurity* (key pair/certificate management environment creation) command. How to use the *soapSetSecurity* command depends on the certification authority to which the application for a certificate is made.

The following shows some examples of using the *soapSetSecurity* command for each certification authority to be used.

Example

If SystemWalker/PkiMGR is the certification authority:

Example 1. Windows

Create a Web service security environment information file and the certificate management file in the certificate management file creation directory by specifying the password for the certificate management file access (Interstage). The installation directory of Interstage is assumed to be "C:\Interstage".

```
soapSetSecurity  -noauth  -f C:\Interstage\F3FMsoap\etc  -p Interstage
```

Example 2. Solaris OE Linux

Create a Web service security environment information file and the certificate management file in the certificate management file creation directory by specifying the password for the certificate management file access (Interstage).

```
soapSetSecurity  -noauth  -f /etc/opt/FJSVsoap/etc  -p Interstage
```

If VeriSign Japan K.K or Japan Certification Services Inc. is the certification authority.

Example 3 Windows

To register the root certificates issued by VeriSign Japan K.K or Japan Certification Services Inc., specify the root certificate list file (C:\Interstage\IS_cert\contractcertlist) in the -rc option of the *soapSetSecurity* command.

```
soapSetSecurity  -noauth  -f C:\Interstage\F3FMsoap\etc  -p Interstage
               -rc C:\Interstage\IS_cert\contractcertlist
```

Example 4 Solaris OE Linux

To register the root certificates issued by VeriSign Japan K.K or Japan Certification Services Inc., specify the root certificate list file (/etc/opt/FJSVisas/contractcertlist) in the -rc option of the *soapSetSecurity* command.

```
soapSetSecurity  -noauth  -f /etc/opt/FJSVsoap/etc  -p Interstage
               -rc /etc/opt/FJSVisas/contractcertlist
```

The following certificates are stored in the certificate management file as the root certificates:

- VeriSign/RSA Secure Server CA
- VeriSign Class 1 Public Primary Certification Authority
- VeriSign Class 2 Public Primary Certification Authority
- VeriSign Class 3 Public Primary Certification Authority
- VeriSign Class 1 Public Primary Certification Authority - G2
- VeriSign Class 2 Public Primary Certification Authority - G2
- VeriSign Class 3 Public Primary Certification Authority - G2
- VeriSign Class 4 Public Primary Certification Authority - G2
- SecureSign RootCA1
- SecureSign RootCA2
- SecureSign RootCA3

Registering Certification Authority Certificates

Register the signer of SOAP digital signatures, receiver of SOAP messages encrypted by the XML encryption, or certificate of the certification authority that issued the site certificate of the SOAP server that conducts SSL-encrypted communication with the certificate management file. Certificates in the format containing certification authority certificates only can be registered using the *soapMngSecurity* (certificate management) command.

Notes

- The same certification authority certificates must be registered on all servers and clients linked for the security function of the Web service.
- If the certification authority is an intermediate certification authority, registration of the certificate must start from the root certification authority.

Example

Register the certification authority certificate with the certificate management file by specifying the alias (cacert).

```
soapMngSecurity  -import  -f  certificate_storage_file_name  -p Interstage
               -alias cacert
```

Registering Site Certificates of the Communication Parties

When encrypting messages using the XML encryption or verifying SOAP digital signatures without site certificate of the signer, it is necessary to obtain the site certificate of the communication party and register it with the certificate management file. Certificates in the format containing only the site certificate of the communication party can be registered using the *soapMngSecurity* (certificate management) command.

Notes

- The certificate of the certification authority that issued the site certificate of the communication party must be registered with the certificate management file.

Example

Register the site certificate of the communication party with the certificate management file by specifying the alias (jiro).

```
soapMngSecurity  -import  -f  certificate_storage_file_name  -p Interstage
               -alias jiro
```

Using a CORBA/SOAP Gateway

If SSL encrypted communication is to be performed in a system environment using a CORBA/SOAP gateway, a Web service security environment and CORBA service SSL environment must be configured.

Using a CORBA/SOAP Server Gateway

The CORBA/SOAP server gateway operates as a Web service RPC server application, and also as a CORBA client application in a server system.

SSL encrypted communication of the Web service can be defined in the Web server SSL environment configuration. From the Interstage Management Console, select [System] > [Services] > [Web Server] > [Web Server Settings], and then [Detailed Settings] and make a definition in "SSL Definition".

For SSL environment configuration for CORBA client applications, refer to "Using CORBA service in SSL mode."

Using a CORBA/SOAP Client Gateway

The CORBA/SOAP client gateway operates as a Web service CORBA server application and also as an RPC client application in a server system.

Refer to Chapter 7 Setting and Use of the Interstage Certificate Environment for SSL environment configuration for the Web service. Refer to Chapter 10 How to Use SSL with the CORBA Service or SSL environment configuration for CORBA client applications. Refer also to *soapgwaddclgw* Command in the Reference Manual (Command Edition).

Chapter 15

User Authentication, SOAP Digital Signature and XML Encryption for Web Services (SOAP)

This chapter explains how to use user authentication, SOAP digital signature, and XML Encryption in the Web services.

Setting User Authentication for SOAP Messages

This section explains the following topics about User Authentication:

- Appending a User Name and a Password to a SOAP Message
- User Authentication for SOAP Messages

Appending a User Name and a Password to a SOAP Message

This section explains how to append a user name and a password, which are used for user authentication, to a SOAP message.

Implementing an Application to Append a User Name and a Password

Before a user name and a password can be appended to a SOAP message, they must be specified in an application, as with basic authentication.

For information on how to specify a user name and a password in the messaging method, refer to “Adding HTTP connection information on proxies, authentication, sessions, etc.” in the “Implementing Applications in the Messaging Method” section of the SOAP Service User’s Guide.

For information on how to specify a user name and a password in the RPC method, refer to “Adding HTTP connection information (stub method) on proxies, authentication, etc.” or “Adding HTTP connection information (DII method) on proxies, authentication, etc.” in the “Implementing Applications in the RPC Method” section of the SOAP Service User’s Guide.

Append a User Name and a Password

Use the Web Service Configuration Edit Tool in a client system environment to configure each SOAP message with an appended user name and password.

To launch the Web Service Configuration Edit Tool, select [Start]-[Programs]-[Interstage]-[SOAP Service]-[Web Service Configuration Edit Tool].

The Web Service Configuration Edit Tool, when launched, displays a window like the one shown in Figure 15-1.

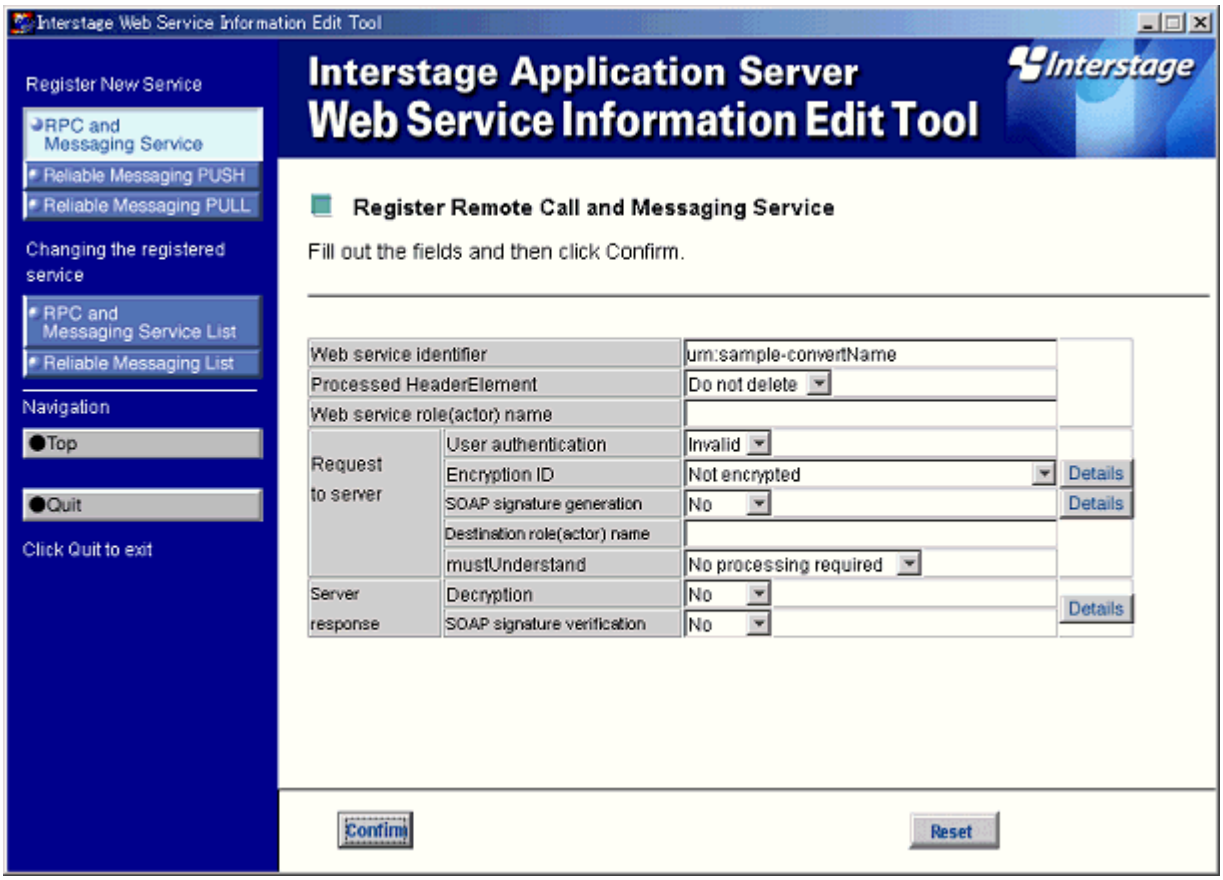


Figure 15-1 Web Service Configuration Edit Tool

- Web service identifier

  Enter the identifier of the Web service.

  For information on how to specify the Web service identifier, refer to Chapter 9 - Managing Web Service Information in the SOAP Service User's Guide.

- Processed header element: Do not delete

  Specify whether to delete the SOAP header element of an incoming response message after its processing has completed.

  The default is “Do not delete.”

- Request transmission setting: User authentication

  Specify whether to append a user name and a password to the SOAP header of each request message when transmitting it.

  The default is “Invalid”

  Select “valid” to append a user name and a password to each SOAP header.

- Request transmission setting: destination role (actor) setting

  Specify the URI of the intermediary for the SOAP message to perform some action on that message.

  If a URI is not specified, the action will be directed at the ultimate destination of the SOAP message.

- Request transmission setting: mustUnderstand

  Specify whether processing of the header elements of the SOAP message by its recipient is mandatory or optional.

  The default is “No need for processing.”

Notes

- If you use the Web Service Configuration Edit Tool to configure SOAP Headers with an appended user name and password, the user name and password that are programmed by an application are automatically appended to the SOAP header of each SOAP message.
- The Windows Start menu may look slightly different depending on the system in use.

User Authentication for SOAP Messages

Implementing an Application that Performs User Authentication on SOAP Messages

An application that performs user authentication on the user name and password appended to each SOAP message is implemented by editing Web service information. There is no need to edit the application program.

Setting User Information

User authentication for SOAP messages involves the use of a certain scheme of user information management at single sign-on. The following information items need to be set to use this scheme:

- Repository server environment setup
- Authentication server environment setup
- Business server environment setup.

Repository Server Environment Setup

The repository server is a server on which user name, password and user role information is managed. For details about how to set up a repository server environment, refer to “Setting up a repository server” in the “Environment Setup (SSO Administrator)” section of the Single Sign-on Guide.

User authentication for SOAP messages does not use the access control information stored on the repository server. The user information that is stored on the repository server must have the user’s role name (ssoRoleName) set in it.

Authentication Server Environment Setup

The authentication server authenticates users by comparing the user name and password appended to each SOAP message against the user information stored on the repository server. For details about how to set up an authentication server environment, refer to “Setting up an authentication server” in the “Environment Setup (SSO Administrator)” section of the Single Sign-on Guide.

Business Server Environment Setup

The server system that implements a Web service to execute user authentication for SOAP messages must have the following single sign-on functions of the business server configured:

- Acquiring the Service ID file
- JAAS (Java (TM) Authentication and Authorization Service) setting

For details about how to get a service ID file, refer to “Getting a service ID file” in “Setting Up an Application Runtime Environment” of "Developing Java Applications" in the “Developing Applications” section of the Single Sign-on Guide.

To use the functionality of JAAS, information on the authentication server and the service ID file must have been set in the login configuration file, which is stored in the following location:

Windows

C:\Interstage\F3FMsoap\conf\issoaplogin.conf

Solaris OE

If the multi-system function is not used or the default system is used
/opt/FJSVsoap/conf/issoaplogin.conf
If a sytem with an extended multi-system function is used
/opt/FJSVsoap/MI/system-name/conf/issoapiogin.conf

Linux

/opt/FJSVsoap/conf/issoaplogin.conf

For a definition of each item of the login configuration file, refer to “Creating a login configuration file” in “Setting Up an Application Runtime Environment” in the “Developing Java Applications” section of the Single Sign-on Guide.

For information on using server authentication or a proxy in SSL communication to gain access to the authentication server at single sign-on, refer to “Running Applications” in the “Developing Java Applications” section of the Single Sign-on Guide.

Notes

- Without the single sign-on function enabled, user authentication for SOAP messages will not function.
- If the system property “java.security.auth.login.config” has been specified at JavaVM launch time, the login configuration file specified in the system property takes precedence.

Setting User Authentication Information

Use the Web Service Configuration Edit Tool in a server system environment to enable user authentication and authorization on the basis of the user name and password appended to each SOAP message.

The Web Service Configuration Edit Tool, when launched, displays a window like the one shown below. A remote call server application window is give here as an example.

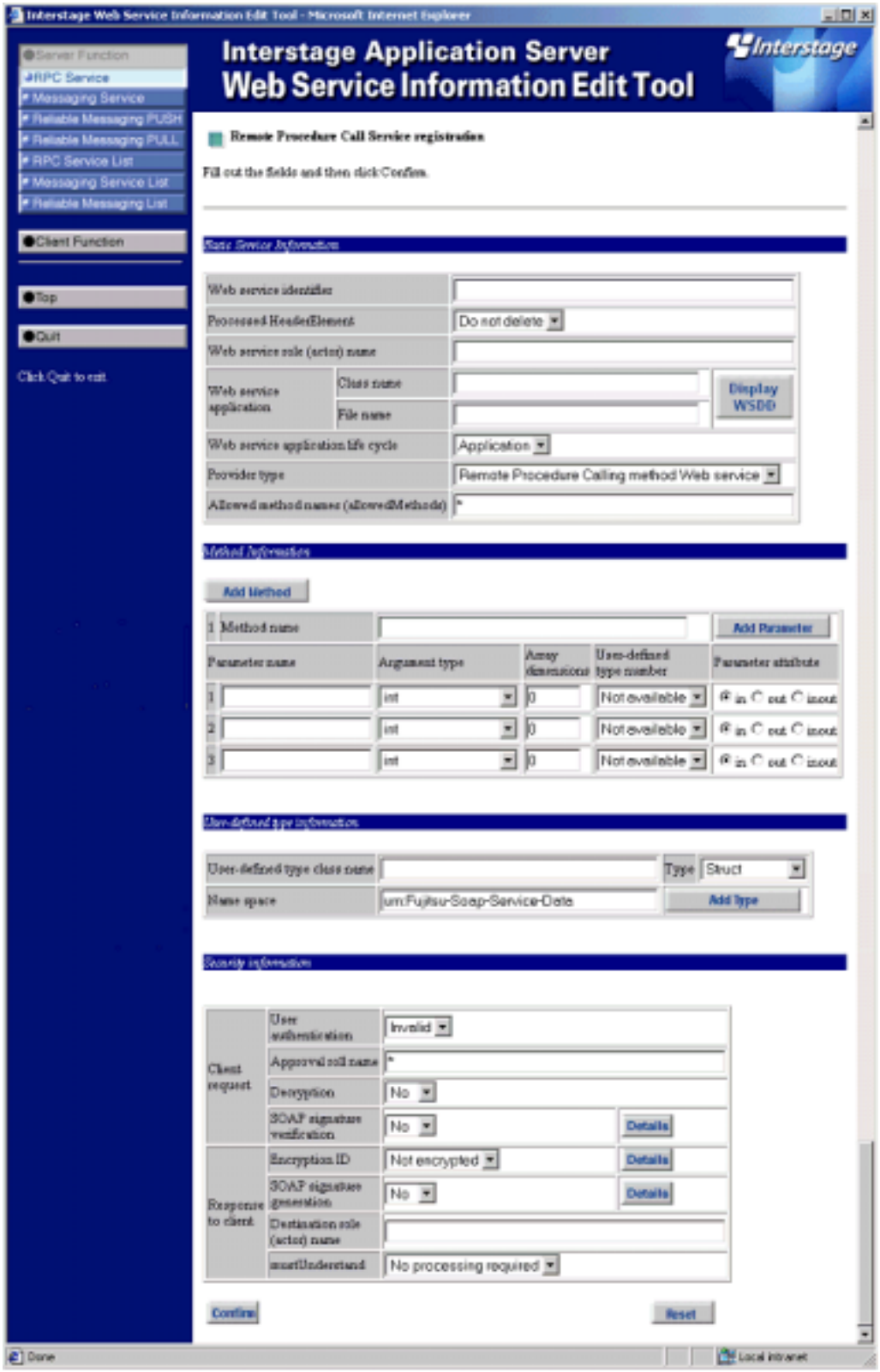


Figure 15-2 Entering User Authentication information

- Web service identifier

  Enter the identifier of the Web service.

  For information on how to specify the Web service identifier, refer to Chapter 9 - Managing Web Service Information in the SOAP Service User's Guide.

- Processed header element: Do not delete

  Specify whether to delete the SOAP header element of an incoming response message after its processing has completed.

  The default is “Do not delete.”

- Web service role (actor) name

  Specify a Web service role (actor) name in URI format.

  Two or more role (actor) names can be entered comma-separated.

  A Web service role (actor) name identifies a Web service role in the path of a transfer defined by SOAP. Only those SOAP header elements having their destination (actor) name matching this value are processed.

  There is no need to specify a role (actor) name in most situations.

- Request transmission setting: User authentication

  Request reception setting: User authentication

  Specify whether to enable user authentication and authorization on the basis of the user name and password appended to each SOAP message. Select “valid” to enable user authentication and authorization.

- Request reception setting: Authorized role name

  List the names of the roles that are authorized to execute the Web service if user authentication o of the user name and password appended to the header of a SOAP message succeeds. Multiple role names can be entered comma-separated.

  The default is “*”, authorizing all roles to execute the Web service.

Notes

- If the user authentication function is enabled with the Web Service Configuration Edit Tool, user authentication executes automatically on the basis of the user name and password appended to the header of a SOAP message.
- If the user authentication function is enabled with the Web Service Configuration Edit Tool when the single sign-on function has not been set, user authentication and authorization would be disabled, with incoming SOAP messages being rejected.

Settings for the SOAP Digital Signature

This section explains the following topics:

- Generating a SOAP Digital Signature
- Verifying the SOAP Digital Signature for SOAP Messages

Generating a SOAP Digital Signature

The following explains the procedure for generating a SOAP digital signature.

Implementing an Application that Attaches a SOAP Digital Signature

An application that generates a SOAP digital signature is implemented by changing the Web service information. There is no need to change the application programs.

For an application that uses the Messaging method, settings for the signature target can be simplified by adding an ID attribute to the element being signed.

In the original assurance function (SOAP digital signature/XML encryption), the following attributes are regarded as an ID attribute:

```
Namespace URI: http://schemas.xmlsoap.org/ws/2002/07/utility
Local Name: “Id”
```

The following shows an example in which an ID attribute with the value "body" is attached to SOAPBody using the SAAJ-API.

Example

```
 .....
String prefix = "wsu";
String uri = "http://schemas.xmlsoap.org/ws/2002/07/utility";
SOAPEnvelope env = ...;
SOAPBody body = env.getBody();
body.addNamespaceDeclaration(prefix, uri);
body.addAttribute(env.createName("Id", prefix, uri), "body");
 .....
```

If an attachment file is to be a target of the SOAP digital signature, the MIME header "Content-Id" needs to be added to the attachment file.

The following shows an example in which the MIME header "Content-Id" with the value "sample.jpg" is added to the attachment file.

Example

```
import javax.xml.soap.*;
  .....
 AttachmentPart _ap = ...;
```

```
_ap.setContentType("image/jpeg");
_ap.setContentId("sample.jpg");
 .....
```

Preparing a Private-key

To generate a SOAP digital signature, a private key and a site certificate corresponding to the private key are needed. For information about how to configure the private key and site certificate, refer to Chapter 14, How to Prepare PKI Environment for Web Services (SOAP).

Note

When acquiring a certificate from the certification authority, a certificate for the digital signature, as for the usage of the key, needs to be acquired.

Settings for the Generation of the SOAP Digital Signature

The Web Service Information Edit Tool is used to set the SOAP digital signature generation.The following figure shows the window that displays detailed information. The following explanation uses the RPC application window as an example.

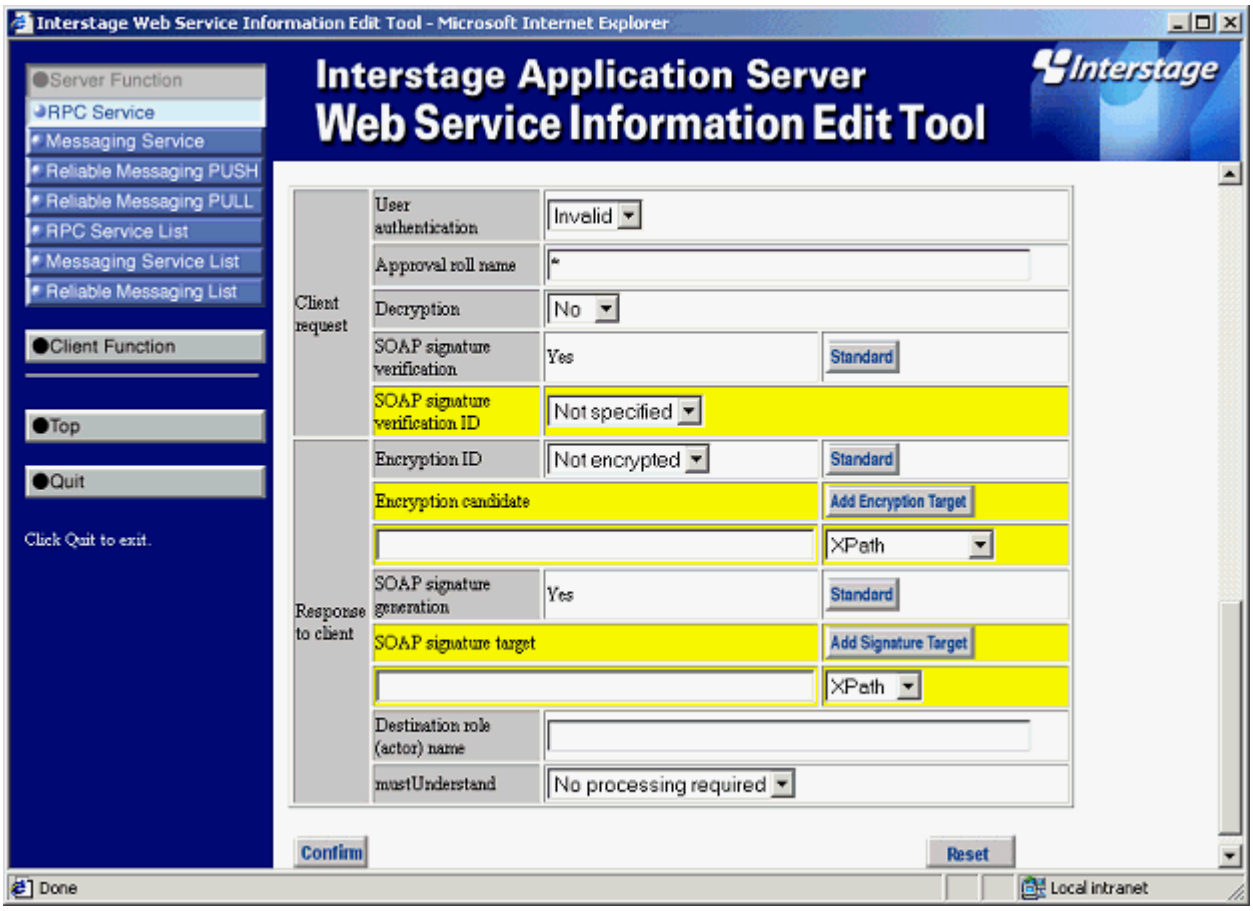


Figure 15-3 Web Service Information Edit Tool – RPC Service – Security Information Screen

- Web service identifier

  Enter the identifier of the Web service.

  For information on how to specify the Web service identifier, refer to Chapter 9 Managing Web Service Information in the SOAP Service User's Guide.

- [Server Function]: Response Sending setup: SOAP signature generation
- [Client Function]: Request Sending setup: SOAP signature generation

  Set whether to generate the SOAP digital signature when sending a request of a SOAP message or a response.

  "No" is set by default.

- [Server Function]: Response Sending setup: **Details** button
- [Client Function]: Request Sending setup: **Details** button

  This button is used to switch between "Details" and "Standard" view of the SOAP digital signature generation setting.

  The above figure shows when the **Details** button has been clicked.

  - SOAP signature target

    Specify the signature targets.

    If the object being signed is the XML element with the ID attribute or the attachment file, select "URI/ID" and then specify the signature target.

    If XPath is to be used to specify the signature target, select "XPath" and then describe the signature target.

    For information about how to specify the signature target, refer to Specifying the Signature Target.

  - Xpath or URI/ID

    Set how to specify the signature target. "XPath" or "URI/ID" can be set. "XPath" is set by default.

  - **Add Signature Target** button

    As many signature targets as desired can be specified. To add two or more signature targets, click the Add Signature Target button to add the next signature target.

- [Server Function]: Request Sending setup: Destination role (actor) name
- [Client Function]: Response Sending setup: Destination role (actor) name

  Specify the URI of the intermediary of the SOAP message if he (she) should perform some operation. Omitting this item will specify the behavior for the ultimate destination of the SOAP message.

- [Server Function]: Request Sending setup: mustUnderstand
- [Client Function]: Response Sending setup: mustUnderstand

  Specify whether the receiver of the SOAP message must always process elements in the header or the processing is selectable.

  "No processing required" is set by default.

Notes

- If the SOAP digital signature generation function is enabled from the Web Service Information Edit Tool, the SOAP digital signature is automatically attached to the SOAP header based on the settings.
- If details are not set, the signature is generated to the results of XML canonicalization after deleting comments from the contents in the SOAP body elements of the SOAP message.
- The settings of the SOAP digital signature generation can also be configured by the soapsecsignconf command. For information on the soapsecsignconf command refer to Reference Manual (Command Edition).

Specifying the Signature Target

The following two types of signature target can be selected for the SOAP digital signature:

- Any nodes inside the SOAP envelope
  - ID Reference
  - XPath filtering
- Attachment files
  - Content-Id

Specifying the Signature Target Using ID Reference

If the SOAP message is created using SAAJ-API, it is possible to add an ID attribute to the element being signed. In the SOAP digital signature, the following attributes are regarded as an ID attribute:

```
Namespace URI: http://schemas.xmlsoap.org/ws/2002/07/utility
Local Name: “Id”
```

To sign an element with an ID attribute, specify a string of the ID attribute value after the number sign ("#") as the signature target.

Example

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body xmlns:wsu="http:// schemas.xmlsoap.org/ws/2002/07/utility"
wsu:Id="body">
   <m:ResponseBody xmlns:m="urn:SampleMsg">
    <Response>response string...</Response>
   </m:ResponseBody>
  </soapenv:Body>
 </soapenv:Envelope
```

If "#body" is specified as the signature target for the above SOAP message, <soapenv:Body> and its contents will be signed.

Specifying the Signature Target Using XPath Filtering

If XPath is specified, nodes for which the result of evaluation of the XPath expression for all nodes inside the SOAP envelope turns out to be true are selected as the signature targets.

Specifying Contents of the SOAP Body as a Signature Target Using XPath Filtering

Specify the signature target as shown below when only the contents in the SOAP body should be signed:

```
ancestor::soapenv:Body
```

Here, "soapenv" represents the namespace prefix of SOAPBody. If the prefix is changed, its value must also be changed when specifying it in XPath.

Specifying Arbitrary Elements as the Signature Target Using XPath Filtering

To sign an element with the namespace URI "urn:sample" and the local name "localName", specify in the following manner:

```
ancestor-or-self::*[local-name()='localName' and namespace-
uri()='urn:sample']
```

Specifying the Signature Target Using Content-Id

To sign an attachment file contained in the SOAP message, specify a string of "cid:" followed by the MIME header "Content-Id" value.

Verifying the SOAP Digital Signature for SOAP Messages

Implementing an Application that Verifies the SOAP Digital Signature

An application that verifies the SOAP digital signature is implemented by changing the Web service information. There is no need to change the application programs.

Preparing a Site Certificate and Certification Authority Certificate

For the verification of a SOAP digital signature, a certificate of the remote site that generates the SOAP digital signature or a certificate of the certification authority that issued the site certificate is needed. For information about how to manage acquired certificates, refer to Chapter 14 How to prepare PKI Environment for Web Services (SOAP).

Settings for the SOAP Digital Signature Verification

The Web Service Information Edit Tool is used to make the settings for the SOAP digital signature verification.

The following figure shows the window that displays detailed information in a server system environment.

The following explanation uses the RPC application window as an example.

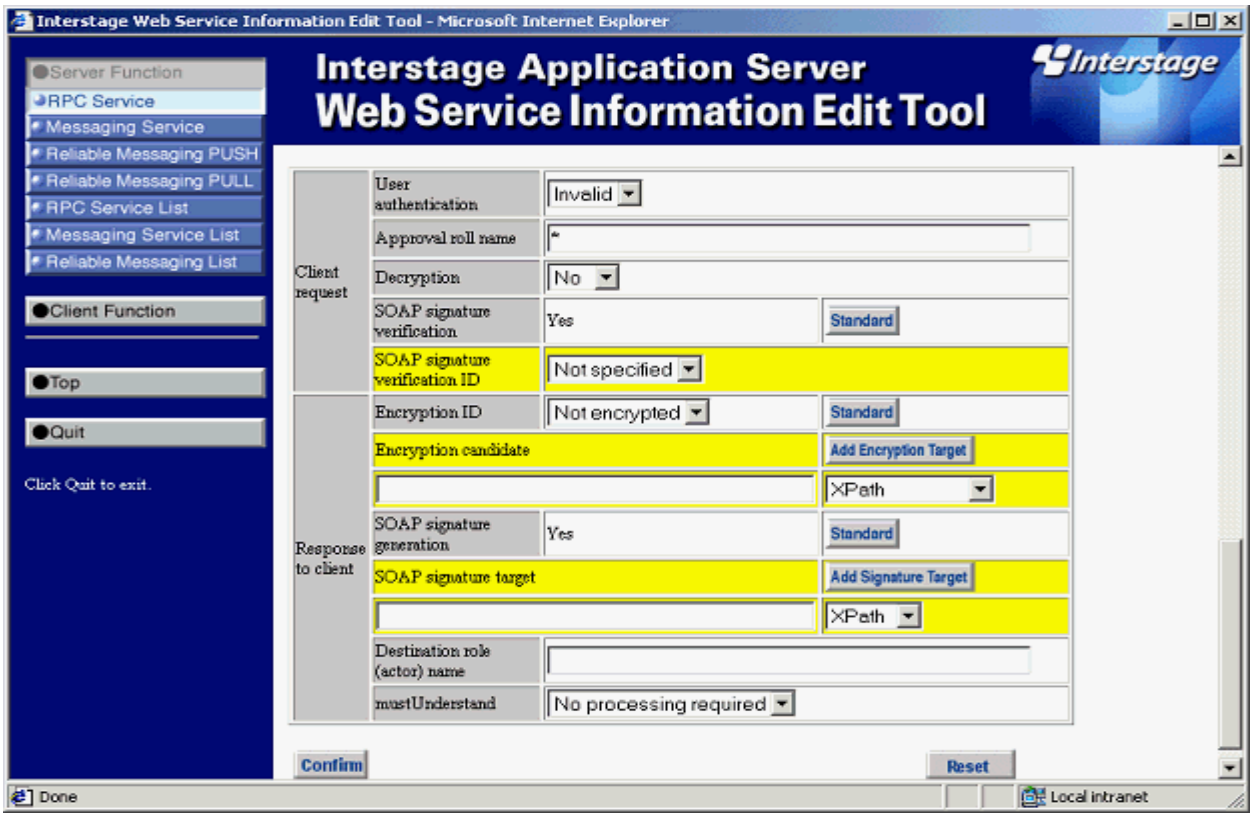


Figure 15-4 Web Service Information Edit Tool – RPC Service – Security Information Screen

- Web service identifier

  Enter the identifier of the Web service.

  For information on how to specify the Web service identifier, refer to the SOAP Service User's Guide.

- Processed HeaderElement: Do not delete

  Set whether to delete processed SOAP header elements after receiving a request/response message.

  "Do not delete" is set by default.

- Web service role (actor) name

  Specify the Web service role (actor) name(s) in URI format.

  When describing two or more role (actor) names, separate them using a comma (",").

  The role (actor) name is a role name of the Web service in the transfer route when SOAP performs the specified transfer. Only the SOAP header elements with the destination role (actor) name matching this value are processed.

  Normally, this item need not be specified.

- [Server Function]: Request Receiving setup: SOAP signature verification

- [Client Function]: Response Receiving setup: SOAP signature verification

  Set whether to verify the SOAP digital signature when receiving a request of a SOAP message or a response.

  "No" is set by default.

- [Server Function]: Request Receiving setup: **Details** button
- [Client Function]: Response Receiving setup: **Details** button

  This button is used to switch between "Details" and "Standard" view of the signature verification setting.

  The above figure shows when the **Details** button has been clicked. (The SOAP signature verification ID is displayed.)

  – SOAP signature verification ID

    This is a detailed setting item when the signature verification is "Yes" upon receipt of a request or a response. Select the alias of the site certificate to be used for verification of the SOAP digital signature from the pull-down menu.

Notes

- If the verification function of the SOAP digital signature is enabled from the Web Service Information Edit Tool, SOAP digital signatures attached to the received SOAP messages are automatically verified based the settings.
- When exchanging a signature between the original assurance functions of the Interstage SOAP Service, there is no need to specify the site certificate (SOAP signature verification ID) to be used for signature verification because the certificate is attached by default.
- The settings of the SOAP digital signature verification can also be configured using the soapsecverifyconf command. For information about the soapsecverifyconf command, refer to the Reference Manual (Command Edition).

Settings for the XML Encryption

This section explains the settings for:

- Encrypting SOAP Messages Using the XML Encryption
- Decrypting SOAP Messages Using the XML Encryption

Encrypting SOAP Messages Using the XML Encryption

The following explains the procedure for encrypting SOAP messages using the XML encryption.

Implementing an Application that Performs Encryption Using the XML Encryption

An application that encrypts the SOAP message using the XML encryption is implemented by changing the Web service information. There is no need to change the application programs.

When creating an application in Messaging method, settings for the encryption target can be simplified by adding an ID attribute to the target element of encryption using the XML encryption.

To encrypt an attachment file using the XML encryption, the MIME header "Content-Id" must be added to the Attachment data.

For information about the addition of ID attributes and how to specify the MIME header "Content-Id," refer to Implementing an Application that Attaches a SOAP Digital Signature

For information about the destination URL when using the XML encryption, refer to Chapter 9 Managing Web Service Information in the SOAP Service User's Guide.

Preparing a Site Certificate

For encryption using the XML encryption, a site certificate of the remote site that decrypts encrypted SOAP messages is needed. For information on how to manage acquired certificates, refer to Chapter 14 How to prepare PKI Environment for Web Services (SOAP).

Settings for Encryption Using the XML Encryption

The Web Service Information Edit Tool is used to make the settings for encryption using the XML encryption.

The following shows the window that displays detailed information in a server system environment.

The following explanation uses the RPC application window as an example.

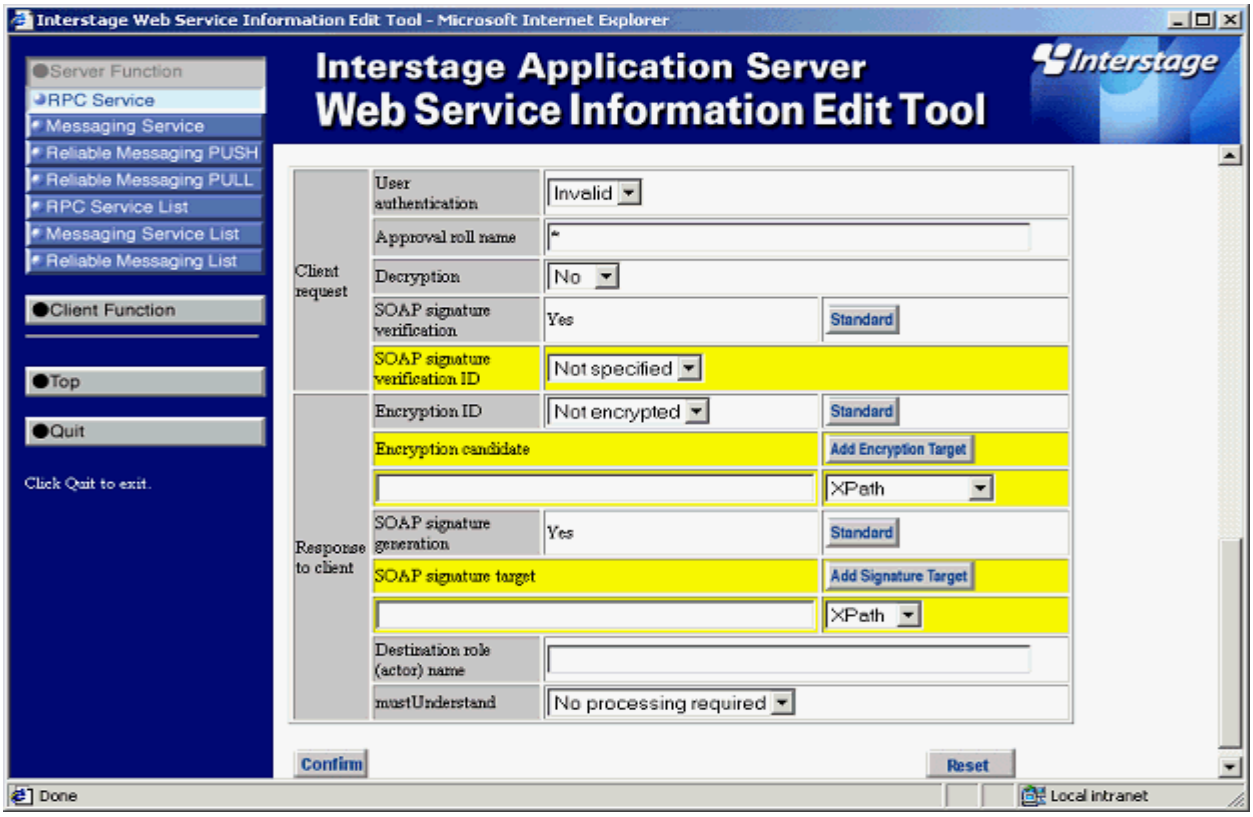


Figure 15-5 Settings for Encryption using the XML Encryption

- Web service identifier

  Enter the identifier of the Web service.

  For information on how to specify the Web service identifier, refer to the SOAP Service User's Guide.

- Processed HeaderElement: Do not delete

  Set whether to delete processed SOAP header elements after receiving a request/response message.

  "Do not delete" is set by default.

- [Server Function]: Response Sending setup: Encryption ID

- [Client Function]: Request Sending setup: Encryption ID

  Make the settings for encryption when sending a SOAP message request/response. Select "Not encrypted" or the alias of the site certificate to be used for encryption using the XML encryption.

  "Not encrypted" is set by default.

- [Server Function]: Response Sending setup: **Details** button

- [Client Function]: Request Sending setup: **Details** button

  This button is used to switch between "Details" and "Standard" view of the encryption setting.

  The above window shows when the **Details** button has been clicked.

  – The Candidate for encryption (optional)

    Specify the target for encryption.

    If an object being encrypted is an element with an ID attribute or an attachment file, select "URI/ID" and then specify the encryption target.

    To specify the encryption target using XPath, select "XPath" or "XPath(Content)" and then specify the encryption target.

    For information on how to specify the encryption target, refer to Specifying the Encryption Target

  – XPath, XPath(Content), or URI/ID

    Set how to specify the encryption target. "XPath", "XPath(Content)", or "URI/ID" can be set. "XPath" is set by default.

- [Server Function]: Request Sending setup: Destination role (actor) name
- [Client Function]: Response Sending setup: Destination role (actor) name

  Specify the URI of the intermediary of the SOAP message if they should perform some operation. Omitting this item will specify the behavior for the ultimate destination of the SOAP message.

- [Server Function]: Request Sending setup: mustUnderstand
- [Client Function]: Response Sending setup: mustUnderstand

  Specify whether the receiver of the SOAP message must always process elements in the header or the processing is selectable.

  "No processing required" is set by default.

Notes

- If the encryption function using the XML encryption is enabled from the Web Service Information Edit Tool, SOAP messages sent based on the settings are automatically encrypted using the XML encryption.
- If no target for encryption is set, the contents of the SOAP body elements in the SOAP messages will be encrypted.
- The settings of the encryption using the XML encryption can also be configured by using the soapsecencconf command. For information on the soapsecencconf command, refer to the Reference Manual (Command Edition).

Specifying the Encryption Target

The following two types of encryption target can be specified for encryption using the XML encryption.

- Any nodes inside the SOAP envelope. (However, the SOAP envelope, SOAP body, and SOAP header are excluded.)
  - ID reference
  - XPath
- Attachment file
  - Content-Id

Specify the Encryption Target using ID Reference

The encryption target can be specified using ID reference in the same manner as the SOAP digital signature using ID reference. For information about how to specify the encryption target, refer to Specifying the Signature Target.

Specify the Encryption Target using XPath

If the encryption target.cannot be specified using ID reference, XPath can be used to specify it. To use XPath to specify the encryption target, it is necessary to specify "XPath" or "XPath(Content)" as the type of encryption target of the Web Service Information Edit Tool.

If "XPath" is specified as the type of encryption target, elements specified by XPath become the encryption target. If, on the other hand, "XPath(Content)" is specified as the type of encryption target contents of the specified elements become the encryption targets.

A SOAP envelope element becomes the initial context node when evaluating an XPath expression.

Example

```
<soapenv:Envelope xmlns:soapenv=”http://schemas.xmlsoap.org/soap/envelope/”
         xmlns:xsd=”http://www.w3.org/2001/XMLSchema”
         xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”>
  <soapenv:Body>
   <m:ResponseBody xmlns:m="urn:SampleMsg">
    <Response>response string...</Response>
   </m:ResponseBody>
  </soapenv:Body>
 </soapenv:Envelope>
```

- descendant::Response

  Specify the Response element, which is a descendant of the SOAP envelope. If "XPath" is specified as the type of encryption, the Response element will be encrypted. If "XPath(Content)" is specified as the type of encryption, the string "response string...," which is the content of the Response element, will be encrypted.

- child::*[local-name()='Body']

  Specify the soapenv:Body element, which is a child of the SOAP envelope. If "XPath(Content)" is specified as the type of encryption, the m:ResponseBody element, which is the content of the soapenv:Body element, will be encrypted. If "XPath" is specified, an error occurs during execution because the SOAP body element itself cannot be encrypted.

- descendant::*[local-name()='ResponseBody' and namespace-uri()='urn:SampleMsg'][1]

  Specify the first m:ResponseBody element, which is a descendant of the SOAP envelope. If "XPath" is specified as the type of encryption, the m:ResponseBody element will be encrypted. If "XPath(Content)" is specified as the type of encryption, the Response element, which is the content of the m:ResponseBody element, will be encrypted.

Specifying the Encryption Target Using Content-Id

The encryption target can be specified using Content-Id in the same manner as the SOAP digital signature using Content-Id. For information about how to specify the encryption target, refer to Specifying the Signature Target.

Decrypting SOAP Messages Using the XML Encryption

The following explains the procedure for decrypting SOAP messages that have been encrypted using the XML encryption.

Implementing an Application that Performs Decryption Using the XML Encryption

An application that decrypts the SOAP message using the XML encryption is implemented by changing the Web service information. There is no need to change the application programs.

Preparing a Private Key

To decrypt an encrypted SOAP message, a private key and a site certificate corresponding to the private key are needed. For information on how to specify the private key and site certificate, refer to Chapter 14 How to prepare PKI Environment for Web Services (SOAP).

Settings for Decryption Using the XML Encryption

The Web Service Information Edit Tool is used to make the settings for decryption using the XML encryption.

The following figure shows the window that displays detailed information in a server system environment.

The following explanation uses the RPC application window as an example.

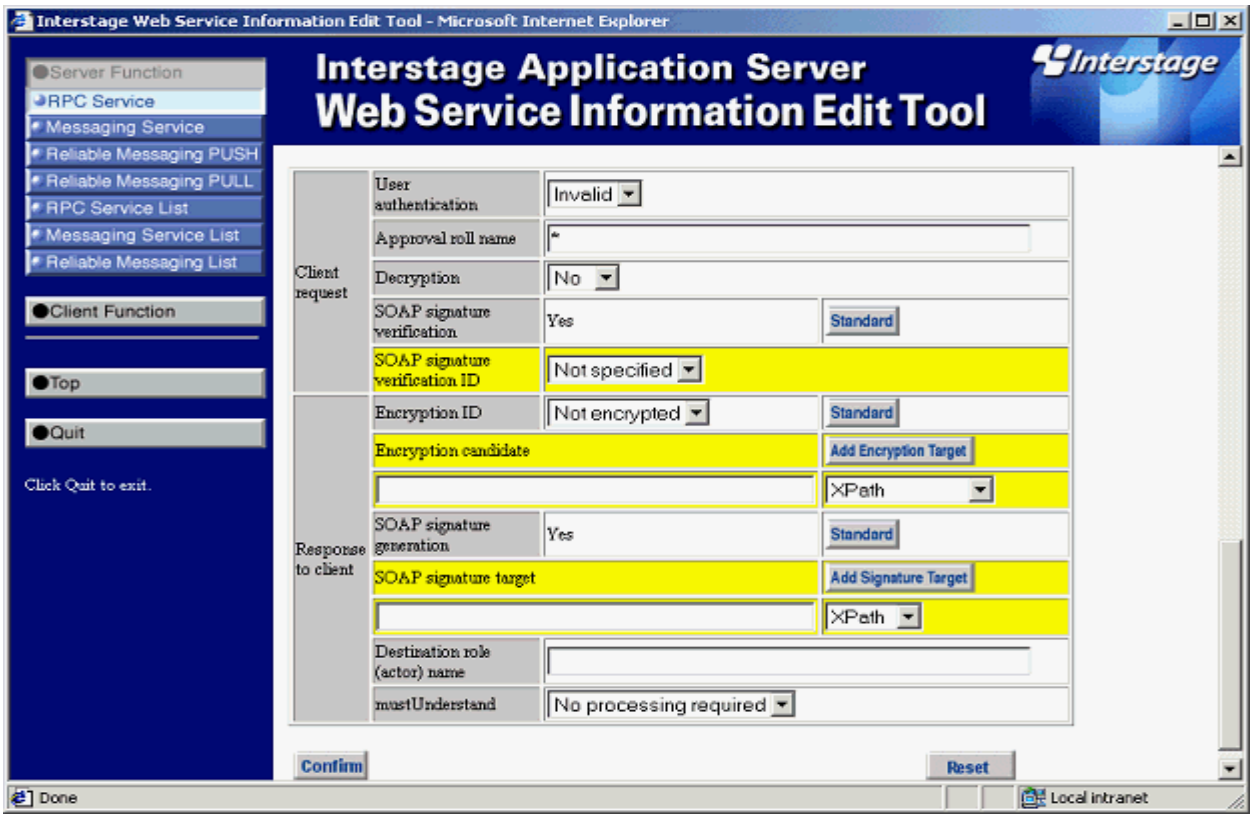


Figure 15-6 Settings for Decryption using the XML encryption

- Web service identifier

  Enter the identifier of the Web service.

  For information on how to specify the Web service identifier, refer to the SOAP Service User's Guide.

- Processed HeaderElement: Do not delete

  Set whether to delete processed SOAP header elements after receiving a request/response message.

  "Do not delete" is set by default.

- Web service role (actor) name

  Specify the Web service role (actor) name(s) in URI format.

  When describing two or more role (actor) names, separate them using a comma (",").

  The role (actor) name is a role name of the Web service in the transfer route when SOAP performs the specified transfer. Only the SOAP header elements with the destination role (actor) name matching this value are processed.

  Normally, this item need not be specified.

- [Server Function]: Request Receiving setup: Decryption
- [Client Function]: Response Receiving setup: Decryption

  Set whether to decrypt the SOAP message using the XML encryption when a request/response message is received.

  "Not encrypted" is set by default.

Notes

- If the decryption function using the XML encryption is enabled from the Web Service Information Edit Tool, a received SOAP message is decrypted automatically using the prepared private key.
- The settings of the decryption using the XML encryption can also be configured using the *soapsecdecconf* command. For information about the *soapsecdecconf* command, refer to the Reference Manual (Command Edition).

Fault Codes

In addition to the faults defined in the “Implementing Messaging Applications” and “Implementing RPC Applications” sections of the SOAP Service User’s Guide, the faults listed below may be encountered while using the user authentication, SOAP e-signature, and XML encryption functions for SOAP messages:

The following fault codes belong to the namespace URI "http://schemas.xmlsoap.org/ws/2002/07/secext".

Table 15-1 Fault Codes

Fault code	Explanation
UnsupportedSecurityToken	An element that cannot be processed by the high-reliability Web service function is contained in the security information attached to the transmitted SOAP message. This fault may occur if a Web service implemented without using the Interstage SOAP Service is connected.
UnsupportedAlgorithm	An algorithm that cannot be processed by the high-reliability Web service function is used for the SOAP digital signature attached to the transmitted SOAP message or encryption using the XML encryption. This fault may occur if a Web service implemented without using the Interstage SOAP Service is connected.
InvalidSecurity InvalidSecurityToken	Since the format of the security information attached to the transmitted SOAP message is invalid, processing by the high-reliability Web service function cannot be performed. This fault may occur if a Web service that does not use the Interstage SOAP Service is connected.
FailedAuthentication	This fault occurs if authentication or permission of the user name or password attached to the SOAP message fails.
FailedCheck	This fault occurs when verification of the SOAP digital signature attached to the SOAP message fails or data encrypted using the XML encryption cannot be decrypted.
SecurityTokenUnavailable	This fault occurs when security information of the site certificate and others to be referenced from the transmitted SOAP message cannot be obtained. This fault may occur if the corresponding security information has been deleted or a Web service that does not use the Interstage SOAP Service is connected.

The following fault code belongs to the namespace URI "http://schemas.xmlsoap.org/ws/2002/07/utility".

Table 15-2 Fault Codes

Fault code	Explanation
MessageExpired	The time stamp attached to the transmitted SOAP message has already expired.

Supported Algorithms

The high-reliability Web service supports the following algorithms.

The namespace prefix "wsse" belongs to the namespace URI "http://schemas.xmlsoap.org/ws/2002/07/security."

Appending a User Name and a Password to a SOAP Message

- Password type
  - wsse:PasswordText

User Authentication for a SOAP Message

- Password type
  - wsse:PasswordText

Generating the SOAP Digital Signature

- Digest algorithm
  - http://www.w3.org/2000/09/xmldsig#sha1
- Signature algorithm
  - http://www.w3.org/2000/09/xmldsig#dsa-sha1
  - http://www.w3.org/2000/09/xmldsig#rsa-sha1
- Canonicalization algorithm
  - http://www.w3.org/2001/10/xml-exc-c14n#
- Transformation algorithm
  - http://www.w3.org/TR/1999/REC-xpath-19991116

Verifying the SOAP Digital Signature

- Digest algorithm
  - http://www.w3.org/2000/09/xmldsig#sha1
- Signature algorithm
  - http://www.w3.org/2000/09/xmldsig#dsa-sha1
  - http://www.w3.org/2000/09/xmldsig#rsa-sha1
- Canonicalization algorithm
  - http://www.w3.org/TR/2001/REC-xml-c14n-20010315
  - http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments
  - http://www.w3.org/2001/10/xml-exc-c14n#
  - http://www.w3.org/2001/10/xml-exc-c14n#WithComments

Notes

The XML canonicalization algorithm that does not remove comments is supported. However, since the format of "#xpointer (xpointer type)" is not supported to specify the signature target, comments cannot be included in the signature target.

- Transformation algorithm
  - http://www.w3.org/2000/09/xmldsig#base64
  - http://www.w3.org/TR/1999/REC-xpath-19991116
  - http://www.w3.org/2000/09/xmldsig#enveloped-signature
  - http://www.w3.org/2002/06/xmldsig-filter2

Encryption using XML Encryption/ Decryption Using XML Encryption

- Symmetric key encryption algorithm
  - http://www.w3.org/2001/04/xmlenc#tripledes-cbc
- Public-key encryption algorithm
  - http://www.w3.org/2001/04/xmlenc#rsa-1_5

Note

In high-reliability functions, because there is no function for saving a symmetric key, the symmetric key encryption algorithm and the public-key encryption algorithm should be used together.

Items Related to WS-Security

- Security token
  - wsse:BinarySecurityToken
  - wsse:UsernameToken
- Encoding method
  - wsse:Base64Binary
  - wsse:HexBinary
- Type supported by wsse:BinarySecurityToken, and wsse:KeyIdentifier
  - wsse:X509v3

Chapter 16

How to Use Reliable Messaging Function for Web Services (SOAP)

This chapter explains how to use the Reliable Messaging function with the Web service.

To use the non-repudiation (signature option) function, the same key pair/certificate environment as that for the SOAP digital signature needs to be constructed. For more information, see “Configuring the Environment to use the Security Function with Web Services (SOAP)” and “Using User Authentication, SOAP E-Signatures, and XML Encryption.”

PUSH Model (Receiving Messages by the Server System)

In the PUSH model, the sender application of the client system sends a SOAP message and the receiver application of the server system receives the SOAP message to process it.

Table 16-1 lists the items that must be agreed upon between the applications.

Table 16-1 PUSH Setting Items

PUSH sender (client system) setting item	PUSH receiver (server system) setting item
Message type ID	Message type ID
Receiver server ID	Receiver server ID
Sender client ID	Sender client ID
Key pair of the sender client	Public key obtained from the sender client
Public key obtained from the receiver server	Key pair of the receiver server
URL with which the receiver server of the PUSH model is registered	URL with which the receiver server of the PUSH model is registered

Each ID must be agreed upon in advance before specifying it for registration of Web service information. To use the non-repudiation (signature option) function, it is necessary to prepare a key pair to be used for SOAP digital signature for each side and to exchange the public keys of the sender client and receiver server in advance using a file.

It is also necessary to place a sender application and a receiver application separately in the receiver server environment and sender client environment respectively and to register Web service information for each application.

Preparing a Key Pair and Public Key Used by the Receiver Server

This section shows the procedure for using the non-repudiation (signature option) function.

The examples shown in this section specify commands used in an Interstage certificate environment. Refer to "Certificate Management" for more information.

For using an old version of certificate environment, refer to "Configuring an Old Certificate Environment or Client System Certificate Environment."

Prepare a key pair for the receiver server. The key pair is the same as that for the SOAP digital signature.

An example of command execution for generating a key pair for the receiving server is shown below:

```
scsmakeenv  -n serverkey  -f filename
```

Next, prepare a public key for the sender client. Since the sender client also needs the public key of the receiver server to use the non-repudiation function, it is necessary to exchange the public key with each other.

The following is an example of command execution for reading a public key file obtained from the sending client:

```
scsenter  -n clientkey  -f clientkeyfile  -p password  -e
```

Note

When exchanging a public key, be sure to use a reliable method such as delivering personally or sending after encryption. If the public keys are not exchanged correctly, the non-repudiation function may not be applicable.

Deploying the Receiver Application

The Web Service Information Edit Tool in the server system environment is used to deploy the receiver application.

For information on how to start the Web service information edit tool, refer to the SOAP Service User 's Guide.

An input window similar to the following is displayed when the Web Service Information Edit Tool is started.

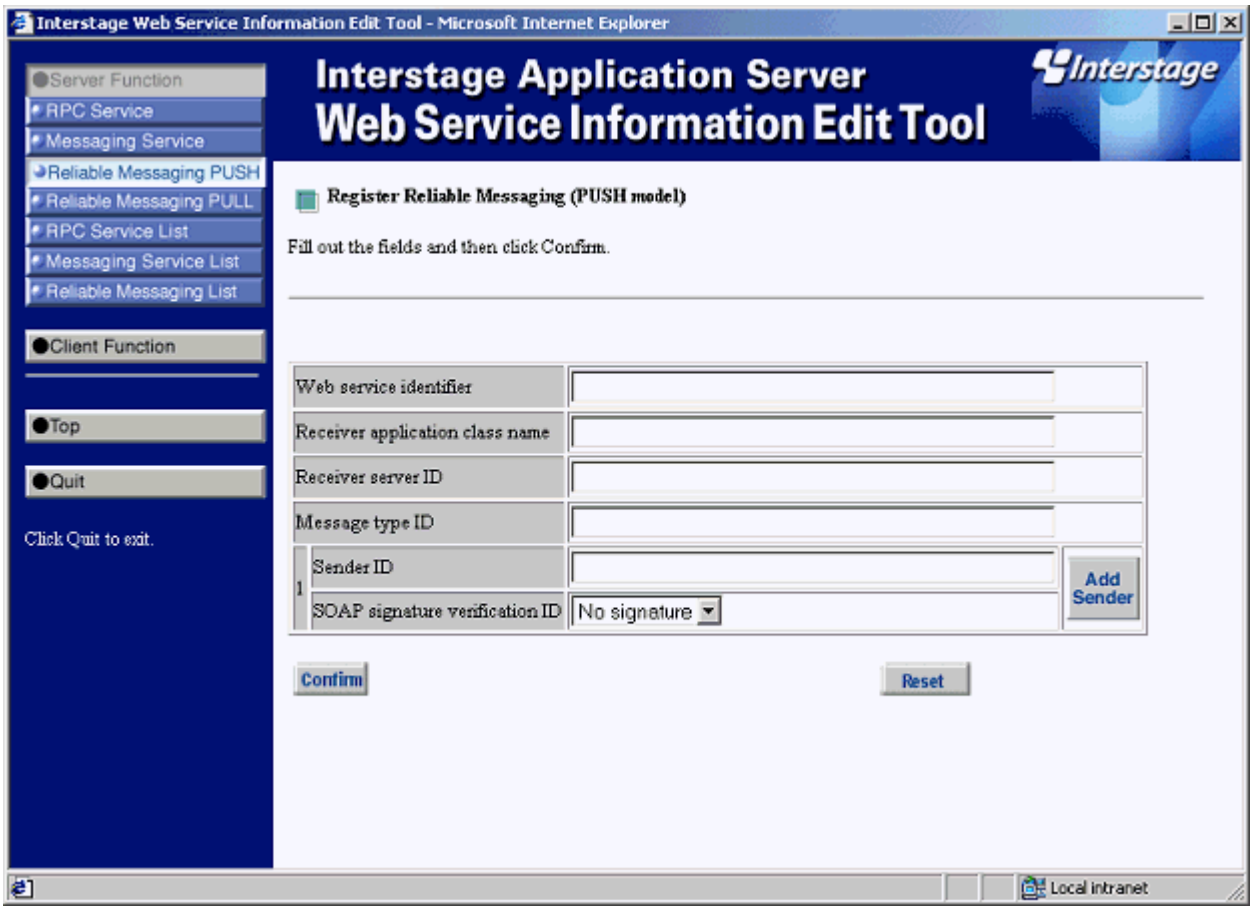


Figure 16-1 Reliable Messaging PUSH Screen - Deploying the Receiver Application dialog

- Web service identifier

  Identifier for the receiver application.

- Receiver application class name

  Specify the Java class name of the receiver application using the full package name.

  The Java class must be placed under a path set to the environmental variable CLASSPATH.

- Receiver server ID

  Specify the ID that represents the receiver server agreed upon with the sender client.

  The format used to specify the Receiver server ID must follow "NMTOKEN" of XML and the characters of the ID must be usable for the file name and directory name. Any name that is unrelated to the machine name and user name is also allowed.

  If the reliable messaging function is to be used in one server system, specify a Receiver server ID that is the same for all SOAP messages and transmission destinations. It is also necessary to specify the same Receiver server ID if multiple servlet containers operate in one server system.

- Message type ID

  Specify the ID that represents the type of message agreed upon with the sender client.

  The format used to specify the message type ID must follow "NMTOKEN" of XML and the characters of the ID must be usable for the file name and directory name. A unique name within one server system must be given.

- Sender ID (Sender client ID)

  Specify the ID for the sender client of the PUSH model agreed upon with the sender client.

  The format used to specify the Sender client ID must follow "NMTOKEN" of XML and the characters of the ID must be usable for the file name and directory name. A unique ID must be given to each message type ID.

- SOAP signature verification ID

  "No Signature" is set by default.

  To use the non-repudiation (signature verification) function, specify an alias of the public key for the sender client. Select the alias specified when the public key obtained from the sender client that was stored in Preparing a Key Pair and Public Key Used by the Receiver Server.

- Add Sender button

  Adds settings of the sender client that sends SOAP messages of the PUSH model of the same message type ID.

  To restrict access to the receiver server, register only one sender client for one message type. For information on access restriction, refer to the SOAP Service User's Guide.

  Clicking the Confirm button at the end enables the Reliable Messaging function, and so the receiver server can now receive reliable messages.

Note

The Receiver server ID, Sender client ID, and Message type ID are used as the directory name and file name for storing reliable messages. Therefore, it is necessary to specify characters that can be used for the directory name and file name in the operating system used. If a character that cannot be used for the directory name or file name is specified, an error occurs when executing the Reliable Messaging function.

Preparing a Key Pair and Public Key Used by the Sender Client

This section shows the procedure for using the non-repudiation (signature option) function.

The examples shown in this section specify commands used in a client package certificate environment. For more information, refer to "Configuring an Old Certificate Environment or Client System Certificate Environment." For using an Interstage certificate environment, refer to "Certificate Management."

Prepare a key pair for the sender client. The key pair is the same as that for the SOAP digital signature.

An example of command execution for generating a key pair for the sender server is shown below:

```
soapSetSecurity -p password -alias clientkey
```

Next, prepare a public key for the receiver client of the PUSH model. Since the receiver server of the PUSH model also needs the public key of the sender client to use the non-repudiation function, it is necessary to exchange the public key with each other.

The following shows an example of command execution to output the public key of the sender client to a file.

```
soapMngSecurity -export -alias clientkey -f clientkeyfile -p password
```

Deliver the file output by this command execution to the receiver server.

The following shows an example of command execution to import the public key file obtained from the receiver server.

```
soapMngSecurity -import -alias serverkey -f serverkeyfile -p password
```

Notes

When exchanging a public key, be sure to use a reliable method (such as delivering personally or sending after encryption) to exchange it. If the public keys are not exchanged correctly, the non-repudiation function may not be applicable.

Deploying the Sender Application

The Web Service Information Edit Tool in the client system environment is used to deploy the sender application.

Start the Web Service Information Edit Tool by selecting
Start | Programs | Interstage | SOAP Service | Web service information edit tool.

An input window similar to the following is displayed when the Web Service Information Edit Tool is started.

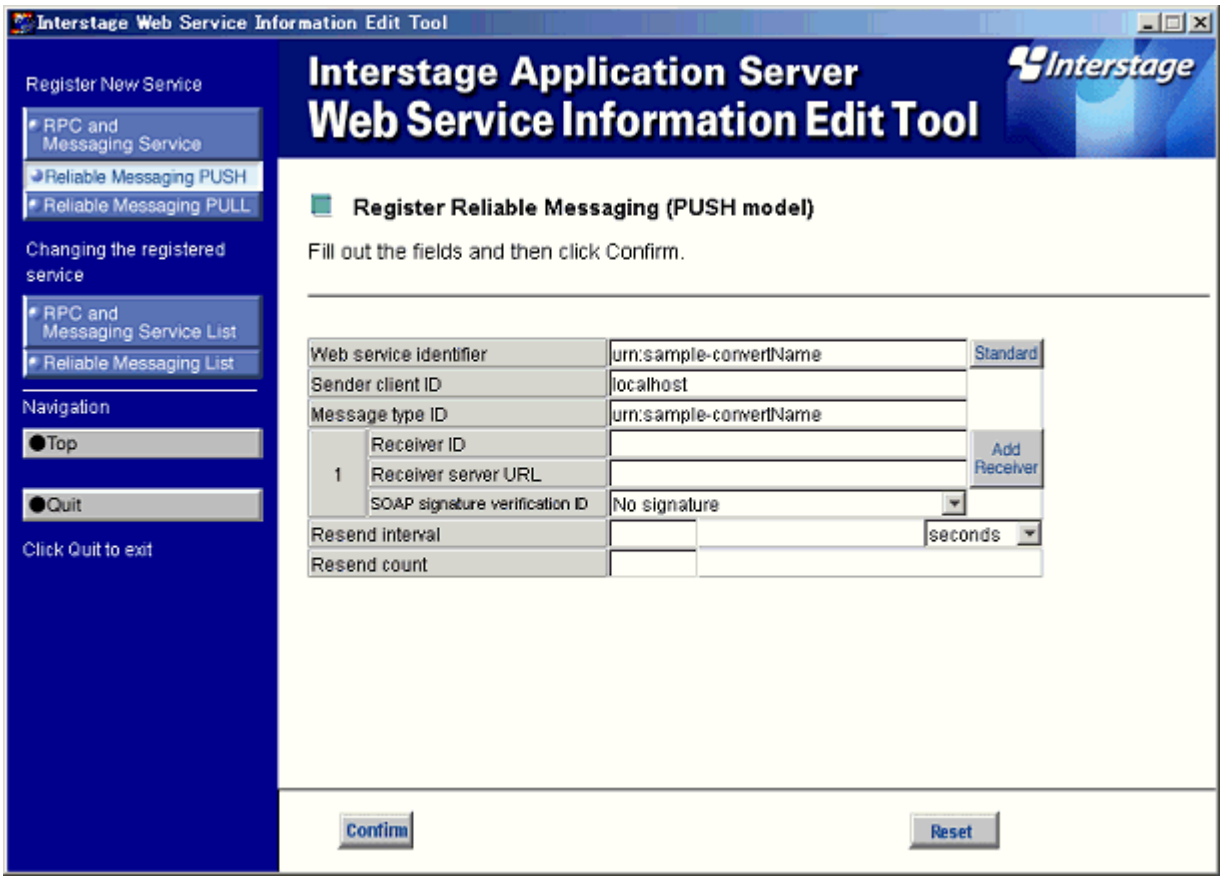


Figure 16-2 Reliable Messaging PUSH Screen - Deploying the Sender Application dialog

- Web service identifier

  Specify the ID that represents the sender client agreed upon with the receiver server.

- Sender client ID

  Specify the ID that represents the sender client agreed upon with the receiver server.

  The format used to specify the Sender client ID must follow "NMTOKEN" of XML and the characters of the ID must be usable for the file name and directory name. Any name that is unrelated to the machine name and user name is also allowed.

  Specify a Sender client ID that is the same to all SOAP messages and transmission destinations in one client system. Also specify the same Sender client ID if multiple JavaVM operate on one client, for example, when multiple Java applications are started.

- Message type ID

  Specify the ID that represents the type of message agreed upon with the receiver server.

  The format used to specify the message type ID must follow "NMTOKEN" of XML and the characters of the ID must be usable for the file name and directory name. A unique name within one client system must be given.

- Receiver ID (Receiver server ID)

  Specify the ID of the receiver server of the PUSH model agreed upon with the receiver server.

  The format used to specify the Receiver server ID must follow "NMTOKEN" of XML and the characters of the ID must be usable for the file name and directory name. A unique ID must be given to each message type ID.

- Receiver server URL (receiver server URL)

  Specify the URL set by the receiver server. Enter the URL corresponding to the Web service identifier specified in Deploying the Receiver Application.

  For details on how to determine the URL, refer to the SOAP Service User's Guide.

- SOAP signature verification ID

  "No Signature" is set by default.

  To use the non-repudiation (signature verification) function, specify an alias of the public key of the receiver server. Select the alias specified when the public key obtained from the receiver server that was stored in Preparing a Key Pair and Public Key Used by the Sender Client.

- Resend interval and Resend count

  Specify the interval and count of retransmission when transmission of a SOAP message fails. If the transmission is not successful within the specified number of times of retransmission, a transmission timeout error of the SOAP message occurs.

- Add Receiver button

  Adds settings of the receiver server that receives SOAP messages from the PUSH model of the same message type ID.

  To restrict access to the receiver server, register only one sender client for one message type ID. For information on access restriction, refer to Chapter 7 How to use the Reliable Messaging Function in the SOAP Service User's Guide.

Clicking the Confirm button at the end enables the reliable messaging function, so the sender client can now send reliable messages.

Notes

The Windows Start menu may be slightly different depending on the system used.

The Receiver server ID, Sender client ID, and Message type ID are used as the directory name and file name for storing reliable messages. Therefore, it is necessary to specify characters that can be used for the directory name and file name in the operating system used. If a character that cannot be used for the directory name or file name is specified, an error occurs when executing the reliable messaging function.

PULL Model (Receiving Messages by the Client System)

In the PULL model, the sender application of the server system (sender server) sends a SOAP message and the receiver application of the client system (receiver client) receives the SOAP message to process it.

Table 16-2 lists the items that must be agreed upon between the applications.

Table 16-2 PULL Model

PUSH sender (client system) setting item	PUSH receiver (server system) setting item
Message type ID	Message type ID
Receiver client ID	Receiver client ID
Sender server ID	Sender server ID
Key pair of the sender server	Public key obtained from the sender server
Public key obtained from the receiver client	Key pair of the receiver client
URL with which the sender server of the PULL model is registered	URL with which the sender server of the PULL model is registered

Each ID must be agreed upon in advance before specifying it for registration of the Web service. To use the non-repudiation (signature option) function, it is necessary to prepare a key pair to be used for SOAP digital signature for each side and to exchange the public keys of the sender server and receiver client in advance using a file.

It is also necessary to place a sender application and a receiver application separately in the sender server environment and receiver client environment respectively and to register Web service information of each application.

Preparing a Key Pair and Public Key Used by the Sender Server

This section shows the procedure for using the non-repudiation (signature option) function.

The examples shown in this section specify commands used in an Interstage certificate environment.

Prepare a key pair for the sender server. The key pair is the same as that for the SOAP digital signature.

An example of command execution for generating a key pair for the sending server is as follows:

```
scsmakeenv  -n serverkey  -f filename
```

Next, prepare a public key for the receiver client. Since the receiver client also needs the public key of the sender server to use the non-repudiation function, it is necessary to exchange the public key with each other.

The following shows an example of command execution to output the public key of the sender server to a file.

```
scsenter  -n clientkey  -f clientkeyfile  -p password  -e
```

Deploying the Sender Application

The Web Service Information Edit Tool in the server system environment is used to deploy the sender application.

For information on how to start the Web service information edit tool, refer to the SOAP Service User 's Guide.

An input window similar to the following is displayed when the Web Service Information Edit Tool is started.

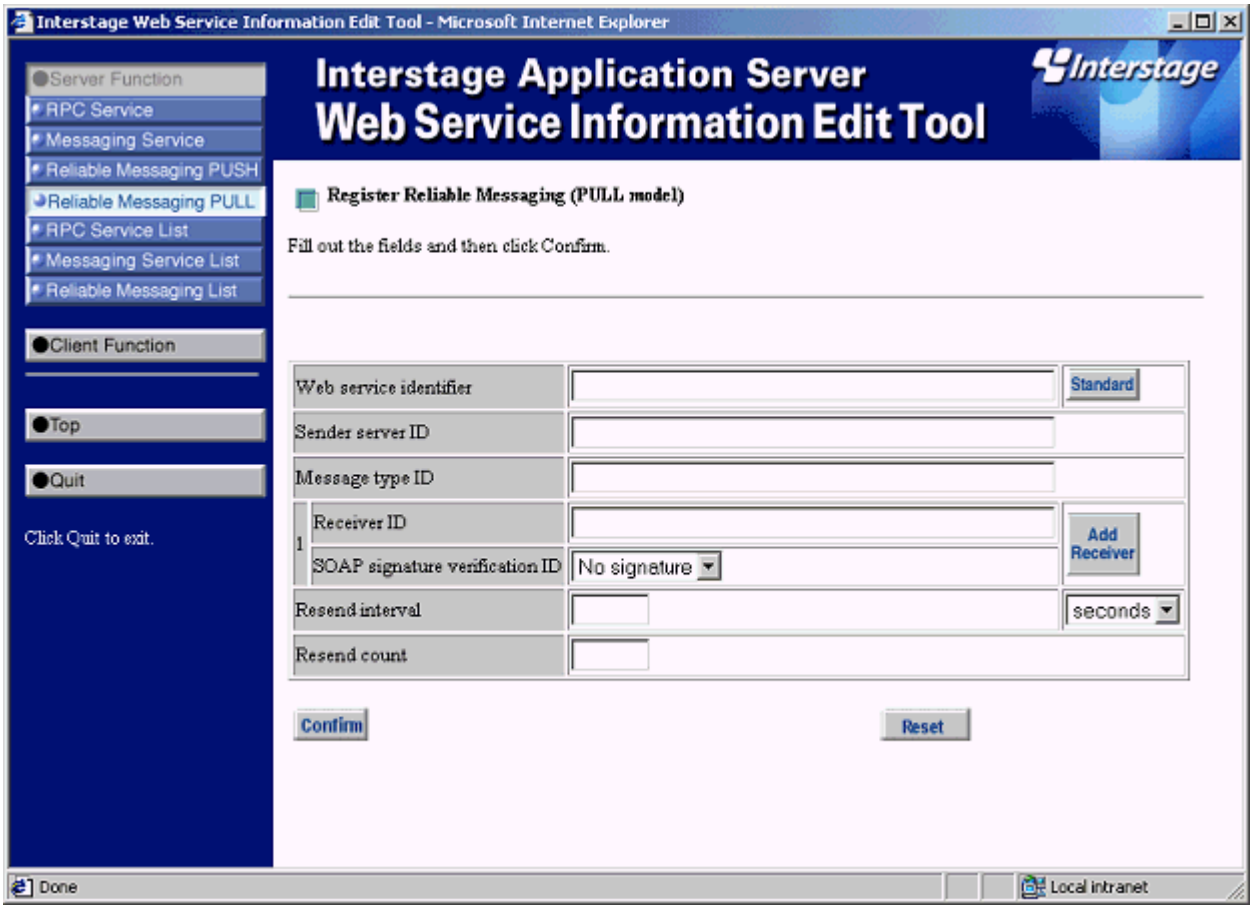


Figure 16-3 Reliable Messaging PULL Screen - Deploying the Sender Application dialog

- Web service identifier

  Identifies the receiver application.

  For details on how to specify the Web service identifier, refer to Chapter 9 Managing Web Service Information in the SOAP Service User's Guide.

- Sender server ID

  Specify the ID that represents the sender server agreed upon with the receiver client.

  The format used to specify the sender server ID must follow "NMTOKEN" of XML and the characters of the ID must be usable for the file name and directory name. Any name that is unrelated to the machine name and user name is also allowed.

  If the reliable messaging function is to be used in one server system, specify a sender server ID that is same to all SOAP messages and transmission destinations.

- Message type ID

  Specify the ID that represents the type of message agreed upon with the receiver client.

  The format used to specify the message type ID must follow "NMTOKEN" of XML and the characters of the ID must be usable for the file name and directory name. A unique name within one server system must be given.

- Receiver ID (Receiver client ID)

  Specify the ID of the receiver client of the PULL model agreed upon with the receiver client.

  The format used to specify the Receiver client ID must follow "NMTOKEN" of XML and the characters of the ID must be usable for the file name and directory name. A unique ID must be given to each message type ID.

- SOAP signature verification ID

  "No Signature" is set by default.

  To use the non-repudiation (signature verification) function, specify an alias of the public key of the receiver client. Select the alias of the public key specified in "Preparing a key pair and public key used by the sender server."

- Add Receiver button

  Adds settings of the receiver client that receives reliable messages of the PULL model for the same message type ID.

  To restrict access to the sender server, register only one receiver client for one message type ID. For information on access restriction, refer to the SOAP Service User's Guide.

- Resend interval and Resend count

  Specify the time during which the SOAP message to be retransmitted should remain ready for sending if transmission of the SOAP message fails. The time interval is calculated by the Resending interval x Resending count.

Click the Confirm button after the input is completed. This enables the reliable messaging function, so the sender server can send reliable messages.

Notes

The Sender server ID, Receiver client ID, and message type ID are used as the directory name and file name for storing reliable messages. Therefore, it is necessary to specify characters that can be used for the directory name and file name in the operating system used. If a character that cannot be used for the directory name or file name is specified, an error occurs when executing the reliable messaging function.

Preparing a Key Pair and Public Key Used by the Receiver Client

This section shows the procedure for using the non-repudiation (signature option) function.

The examples shown in this section specify commands used in a client package certificate environment.

Prepare a key pair for the receiver client. The key pair is the same as that for the SOAP digital signature.

The following shows an example of command execution to generate a key pair for the receiver client.

```
soapSetSecurity -p password -alias clientkey
```

Next, prepare a public key for the sender server of the PULL model. Since the sender server also needs the public key of the receiver client to use the non-repudiation function, it is necessary to exchange the public key with each other.

The following shows an example of command execution to output the public key of the receiver client to a file.

```
soapMngSecurity -export -alias clientkey -f clientkeyfile -p password
```

Deliver the file output by this command execution to the receiver server.

The following shows an example of command execution to import the public key file obtained from the sender server.

```
soapMngSecurity -import -alias serverkey -f serverkeyfile -p password
```

Deploying the Receiver Application

The Web Service Information Edit Tool in the client system environment is used to deploy the receiver application.

Start the Web Service Information Edit Tool by selecting Start | Programs | Interstage | SOAP Service | Web service information edit tool.

An input window similar to the following is displayed when the Web Service Information Edit Tool is started

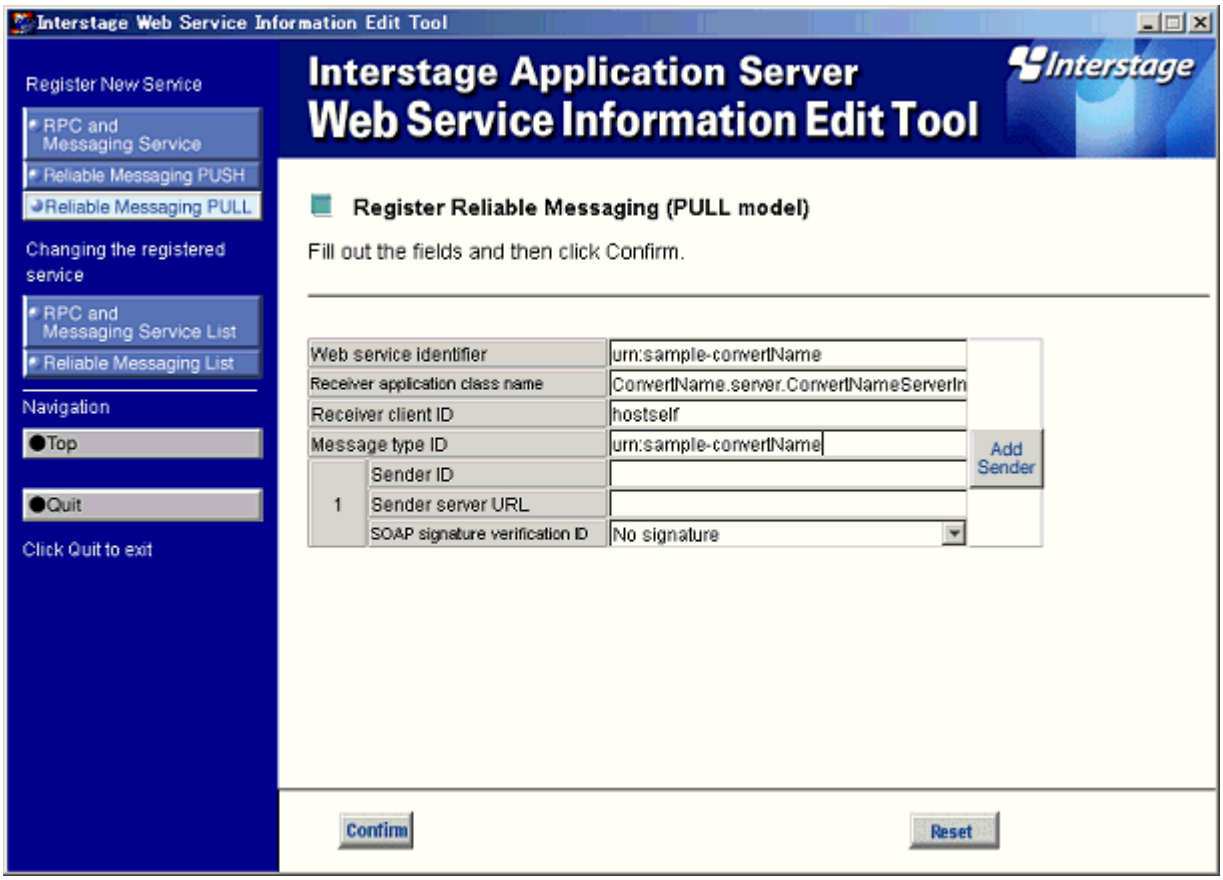


Figure 16-4 Reliable Messaging PULL Screen - Deploying the Receiver Application dialog

- Web service identifier

  Name to identify the deployed Web service information.

  This name must be unique in the system.

- Receiver application class name

  Specify the Java class name of the receiver application using the full package name.

- The Java class must be placed under a path set to the environmental variable CLASSPATH.Receiver client ID

  Specify the ID that represents the receiver client agreed upon with the sender server.

  The format used to specify the Receiver client ID must follow "NMTOKEN" of XML and the characters of the ID must be usable for the file name and directory name. Any name that is unrelated to the server name and user name is also allowed.

  Specify a Receiver client ID that is same to all SOAP messages and transmission destinations in one client system. Also specify the same Receiver client ID if multiple JavaVM operate in one client system, for example, when multiple Java applications are started.

- Message type ID

  Specify the ID that represents the type of message agreed upon with the sender server.

  The format used to specify the message type ID must follow "NMTOKEN" of XML and the characters of the ID must be usable for the file name and directory name. A unique name within one client system must be given.

- Sender ID (Sender client ID)

  Specify the ID of the sender server of the PULL model agreed upon with the sender server.

  The format used to specify the Sender server ID must follow "NMTOKEN" of XML and the characters of the ID must be usable for the file name and directory name. A unique ID must be given to each message type ID.

- Sender server URL (sender server URL)

  URL set by the sender server. Enter the URL corresponding to the Web service identifier specified in Deploying the Sender Application.

  For details on how to determine the URL, refer to the SOAP Service User's Guide.

- SOAP signature verification ID

  "No Signature" is set by default.

  To use the non-repudiation (signature verification) function, specify an alias of the public key of the sender server. Select the alias specified when the public key obtained from the sender server that was stored in Preparing a Key Pair and Public Key Used by the Receiver Client.

- Add Sender button

  Adds settings of the sender server that sends SOAP messages of the PULL model of the same message type ID.

  To restrict access to the sender server, only register one receiver client for one message type ID. For information about access restriction, refer to the SOAP Service User's Guide.

Click the Confirm button after the input is complete. This enables the reliable messaging function, so the receiver client can receive reliable messages.

Notes

The Windows Start menu may be slightly different depending on the system used.

The Sender server ID, Receiver client ID, and message type ID are used as the directory name and file name for storing reliable messages. Therefore, it is necessary to specify characters that can be used for the directory name and file name in the operating system used. If a character that cannot be used for the directory name or file name is specified, an error occurs when executing the reliable messaging function.

Part VI
Security Systems for the ebXML Message Service

The ebXML Message Service can be used with the following Windows(R) system or Solaris OE systemproducts:

- Interstage Application Server Enterprise Edition

Chapter 17
How to use SSL with the ebXML Message Service

This chapter explains how to use SSL with the ebXML Message Service.

With the ebXML Message Service, you can use the following two environments that manage certificates and private-keys required for encryption and signature processing:

- Interstage certificate environment
- Former certificate/key management environment

Configure whichever of the above two environments is appropriate.

Interstage Certificate Environment

To use an Interstage certificate environment, configure an SSL environment by referring to Configuring and using an Interstage certificate environment. Then, set up the environment so that the SSL function can be used from the ebXML Message Service by referring to the ebXML Message Service User's Guide.

Former Certificate/Key Management Environment

To use a former certificate/key management environment, configure a former certificate/key management environment for the SOAP service by referring to Setting up an environment for using security functions from the Web service (SOAP).

Chapter 18

How to use XML Digital Signature with ebXML Message Service

This chapter explains how to use the XML digital signature with the ebXML Message Service.

With the ebXML Message Service, you can use the following two environments that manage certificates and private-keys required for encryption and signature processing:

- Interstage Certificate Environment
- Former Certificate/Key Management Environment

Configure whichever of the above two environments is appropriate.

Interstage Certificate Environment

To use an Interstage certificate environment, configure an SSL environment by referring to "Setting and Use of the Interstage Certificate Environment". Then, set up the environment so that the SSL function can be used from the ebXML Message Service by referring to the ebXML Message Service User's Guide.

Former Certificate/Key Management Environment

To use a former certificate/key management environment, configure a former certificate/key management environment for the SOAP service by referring to "Related Function Environments" in "ebXML Message Service User's Guide"..

Index

access control, B-4

Acquiring and Registering Certificates (for both the Server and Client), 10-4

Append a User Name and a Password, 15-2

Appending a User Name and a Password to a SOAP Message, 15-2

Authentication
  User, 3-2

Authentication object log
  tracing, B-15

authentication objects, B-2

backup
  security, 2-2

CA (Certification Authority), 7-4

certificate
  acquiring, 8-9
  registering, 8-10

certificate issuance
  requesting, 7-10

certificate management, 7-19

Certificate Management, 7-19

certificate signing request (CSR)
  creating, 7-9

certificate/key management environment, 8-3
  creating, 8-7
  environment setting, 8-5
  managing, 8-14
  SSL libraries used with, 8-2

certificates, 7-2
  defining use of, 7-17

certificates and CRL
  registering, 7-10

Certificates and Private Keys, 7-2

client certificate
  obtaining, 8-12
  operating, 8-11
  registering, 8-12

Component Transaction Service, A-9
  security, B-14

Component Transaction Service Security
  access control, B-7
  authentication object, B-5
  overview, B-5
  user authentication, B-5
  user registration, B-6
  using Infordirectory Server, B-5

config file
  editing, 10-8

Configuration Management function
  possible security risks, 1-42
  resources to be protected, 1-41
  security, 1-41
  threat countermeasures, 1-42
  usage model, 1-41

Configuration Model, 1-28

configuring
  Interstage certificate environment, 7-5

Configuring an Interstage Certificate Environment and Creating a Certificate Signing Request (CSR), 7-9

Configuring Certificate Settings, 7-17

Configuring Environments, 7-5

Configuring the Interstage Certificate Environment, 7-13

Configuring the Interstage Certificate Environment with CSR, 7-8

Configuring the Interstage Certificate Environment with PKCS#12, 7-13

Constructing SSL Linkage Environment, 10-4

CORBA server environment setup, 10-6

CORBA Server Environment Setup, 10-6

CORBA Service
- communication data, 2-19
- Java applet creation and operation, 2-19
- port number, 2-19
- security measures, 2-18
- SSL linkage, 10-3
- unauthorized resource file access, 2-18

Countermeasures Against Security Risks, 1-22

Countermeasures Against Threats, 1-3, 1-16, 1-39

creating
- certificate signing request (CSR), 7-9

Creating entries, 3-18

CRL
- registering, 8-10

Database Linkage Service, 1-12
- periodic backup, 1-18
- resources protected, 1-12
- security, 1-12
- security risks, 1-15
- threat countermeasures, 1-16
- using resource security function, 1-18

Decrypting SOAP Messages Using the XML Encryption, 15-20

defining
- use of certificates, 7-17

Defining a Private Key/Certificate in CORBA Service, 10-7

Defining the Use of Certificates, 7-17

Deploying the Receiver Application, 16-3, 16-12

Deploying the Sender Application, 16-6, 16-10

Developing the CORBA Application, 10-3

Editing config File, 10-8

EJB Service
- periodic backup, 2-14
- resources protected, 2-12
- security measures, 2-12
- security risks, 2-13
- SSL encryption, 2-14
- threat countermeasures, 2-13

ejbchangemode command, A-19

Encrypting SOAP Messages Using the XML Encryption, 15-16

encryption communication
- setting/unsetting, 11-3

Environment Setup for Event Service, 10-9

Event Service
- environment setup, 10-9
- security measures, 2-22

Executing the Security-Enhancing Environment Setup Command, A-10

Fault Codes, 15-23

For Dynamic Generation and Operation (for Environment Setting using the Event Service Operation Command), 10-10

For Dynamic Generation and Operation (for Environment Setting using the Interstage Integration Command), 10-9

For Static Generation and Operation, 10-9

Fujitsu Enabler
- account for Fujitsu Enabler, 2-25

Generating a SOAP Digital Signature, 15-9

Generating an Extended System, A-10

Generating an Interstage System Definition File, A-10

How to Use SSL with the CORBA Service, 10-1

HTML Page Editing Service
- customize, B-15

HTTP Data Communication Using HTTP Tunneling, 4-2

HTTP proxy server
- settings, 4-13

HTTP tunneling, 4-4
- Java applets, 4-10

parameters, 4-8
setup, 4-8
writing HTML, 4-7

HTTP Tunneling Setup, 4-4

HTTP-IIOP gateway, 4-4

IJServer
execution, 2-23
security measures, 2-23
unauthorized resource file access, 2-23

Implementing an Application that Attaches a SOAP Digital Signature, 15-9

Implementing an Application that Performs Decryption Using the XML Encryption, 15-20

Implementing an Application that Performs Encryption Using the XML Encryption, 15-16

Implementing an Application that Performs User Authentication on SOAP Messages, 15-4

Implementing an Application that Verifies the SOAP Digital Signature, 15-13

Implementing an Application to Append a User Name and a Password, 15-2

Infodirectory
user registration, B-6

InfoDirectory, B-14
API object wrapping, B-2
running Interstage, B-15

Infodirectory Server
authentication requests, B-7
operational design, B-5

InfoProvider Pro
content permissions, 2-9
environment definition file permissions, 2-9
security measures, 2-9
user authentication, 2-9

Initializing Interstage, A-10

Interstage
access control, B-4
user authentication, B-2

Interstage certificate environment
configuring, 7-5
setting and use of, 7-1

Interstage HTTP Server
communication data, 2-4
DoS attacks, 2-4
HTTP trace method exploitation, 2-6
password information linkage, 2-5
security measures, 2-4
unauthorized resource file access, 2-5

Interstage installation resources
security, 2-2

Interstage JMS
environment setup, 11-4
security measures, 2-17
unauthorized resource file access, 2-17

Interstage Management Console
resources protected, 1-2
security, 1-2
security risks, 1-3
threat countermeasures, 1-3

Interstage Management Console and Interstage Operation Tool, 1-2

Interstage Operation Tool
communication data, 2-3
environment definition file permissions, 2-3
resources protected, 1-2
security, 1-2
security measures, 2-3
security risks, 1-3
threat countermeasures, 1-3
user accounts, 2-3

Interstage Single Sign-on, 1-28
applying patches, 1-34
configuration model, 1-28
operating and managing business server, 1-33
security, 1-28
security risks, 1-29
threat countermeasures, 1-30

IP access control, 3-3

J2EE Application, 1-5

J2EE applications
resources protected, 1-5
security, 1-5
security risks, 1-7

threat countermeasures, 1-9

J2EE deployment tool
security measures, 2-15
unauthorized resource file access, 2-15

J2EE resource access definition
password information leakage, 2-16
security measures, 2-16

Java applets
HTTP tunneling, 4-10

manager DN, B-14

Mechanism of the SSL Linkage Function, 10-3

Multi Server Management
configuration model, 1-36
resources protected, 1-38
security, 1-35
security risks, 1-38
threat countermeasures, 1-39

Notes, A-11, A-28

OLTP function
resources protected, 1-19
security, 1-19
security risks, 1-21
threat countermeasures, 1-22

OLTP Function, 1-19

Online collation, 3-4

Operating the SSL Linkage, 10-5

output messages, A-21

overview
HTTP tunneling, 4-4

Portable-ORB
communication data, 2-20
Java applet creation and operation, 2-21
security measures, 2-20
unauthorized resource file access, 2-20, 2-22

Possible Countermeasures, 1-9

Possible Security Risks, 1-7

Possible Security Risks to Resources, 1-3, 1-38

Possible Threats, 1-29

Possible Threats to Resources, 1-15, 1-21

Potential Security Threats, 1-25

Preparing a Key Pair and Public Key Used by the Receiver Client, 16-12

Preparing a Key Pair and Public Key Used by the Receiver Server, 16-2

Preparing a Key Pair and Public Key Used by the Sender Client, 16-6

Preparing a Key Pair and Public Key Used by the Sender Server, 16-9

Preparing a Private Key, 15-20

Preparing a Private-key, 15-10

Preparing a Site Certificate, 15-16

Preparing a Site Certificate and Certification Authority Certificate, 15-13

private key
creating, 8-9

private key/certificate
defining in CORBA Service, 10-7

private keys, 7-2

PULL Model (Receiving Messages by the Client System), 16-9

PUSH Model (Receiving Messages by the Server System), 16-2

registering
certificates and CRL, 7-10

Registering Certificates and CRL, 7-10

Registering PKCS#12 Data, Certificates, and CRLs, 7-14

registering resources, 8-12

Registering the Interstage System Definition File, A-10

Registering the User PIN, 9-3

Relating Directives, 3-7, 3-12, 3-28

requesting
certificate issuance, 7-10

Requesting Certificate Issuance, 7-10

resource registration, 8-12

resources

CORBA Service, A-4
EJB Service, A-16
EJB Service operation, A-18
environment, A-17
environment construction, A-16, A-17, A-25
environment setup for Interstage resources protection, A-4
Interstage JMS, A-25
protecting Interstage resources, A-2

Resources Requiring Security Protection, 1-24

Resources to be Protected, 1-2, 1-5, 1-12, 1-19, 1-38

security
Component Transaction Service, B-14
Database Linkage Service, 1-12
Interstage Management Console and Interstage Operation Tool, 1-2
Interstage Single Sign-on, 1-28
J2EE applications, 1-5
measures, 2-1
OLTP function, 1-19
risks, 1-1
Smart Repository, 1-24
web services, 1-11

Security Design
Component Transaction Service, B-5

security functions
access control, B-4
user authentication, B-2

security measures
backup, 2-2
CORBA Service, 2-18
EJB Service, 2-12
Event Service, 2-22
IJServer, 2-23
InfoProvider Pro, 2-9
Interstage HTTP Server, 2-4
Interstage installation resources, 2-2
Interstage JMS, 2-17
Interstage Operation Tool, 2-3
J2EE deployment tool, 2-15
J2EE resource access definition, 2-16
notes on user accounts, 2-2
Portable-ORB, 2-20
Servlet Service, 2-10
Smart Repository, 2-24

Security Measures, 1-30

service environments
setting up various, 7-17

Servlet Service
communication data, 2-11
environment setup, 11-2
security measures, 2-10
session usage, 2-10
web application deployment, 2-10
web application development, 2-10
web application root directory, 2-10

Set the Interstage HTTP Server Environment Definition File, 3-20

setting
Interstage certificate environment, 7-1
various service environments, 7-17

Setting and Registering the SSL Environment with the CORBA Service (for both the Server and Client), 10-4

Setting the Directory Server Environment, 3-18

Setting the IP Access Control, 3-11

Setting the Online Collation Function, 3-15

Setting the SSL Information in the CORBA Application (Server Application Only), 10-4

Setting the User Authentication, 3-5

Setting to be Made When an HTTP Proxy Server is to be Used, 4-13

Setting up Access Permissions in the Interstage Certificate Environment, 7-6

Setting up HTTP Tunneling, 4-8

Setting up the Environment Definition File, 9-4

Setting Up Various Service Environments, 7-17

Setting User Authentication, 15-2

Setting User Authentication Information, 15-6

Setting User Information, 15-4

Settings for Decryption Using the XML Encryption, 15-21

Settings for Encryption Using the XML Encryption, 15-16

Settings for the Generation of the SOAP Digital Signature, 15-10

Settings for the SOAP Digital Signature, 15-9

Settings for the SOAP Digital Signature Verification, 15-13

Settings for the XML Encryption, 15-16

Smart Repository, 1-24
- accessing Smart Repository server, 2-24
- blocking external access, 2-24
- operation, 2-24
- periodic data backup, 1-27
- resources protected, 1-24
- restriction of services, 2-24
- security, 1-24
- security measures, 2-24
- security risks, 1-25
- threat countermeasures, 1-25

SMEE libraries
- types of, 8-2

Specifying the Addition of SSL Information at Definition of Object Reference, 10-6

SSL
- environment setup in client, 10-7
- using with J2EE, 11-1

SSL Environment Setup in Client, 10-7

SSL information
- specifying at object reference definition, 10-6

SSL libraries
- used with certificate/key management environment, 8-2

SSL linkage
- CORBA Service, 10-3

SSL Linkage in the IPv6 Environment, 10-5

SSL Linkage of the CORBA Service, 10-3

Supported Algorithms, 15-25

tdresetaso command
- synchronizing information, B-15

tdsetauthmanager command
- set DN, B-14

Threats and Security Measures, 1-25

Types of Authentication, 3-2

user accounts
- security, 2-2

user authentication, B-2
- with authentication objects, B-2
- with Web Server Functions, B-3

User Authentication, 3-2, B-2

User Authentication for SOAP Messages, 15-4

User Authentication with Authentication Objects, B-2

User Authentication with Web Server Functions, B-3

User Authentication, SOAP Digital Signature and XML Encryption for Web Services (SOAP), 15-1

Verifying the SOAP Digital Signature for SOAP Messages, 15-13

Web server definitions
- customize, B-15

Web server environment
- setup, 4-4

Web Server Functions, B-3

Web server online reference function
- linkage to, B-15

web services
- security, 1-11

Web Services, 1-11